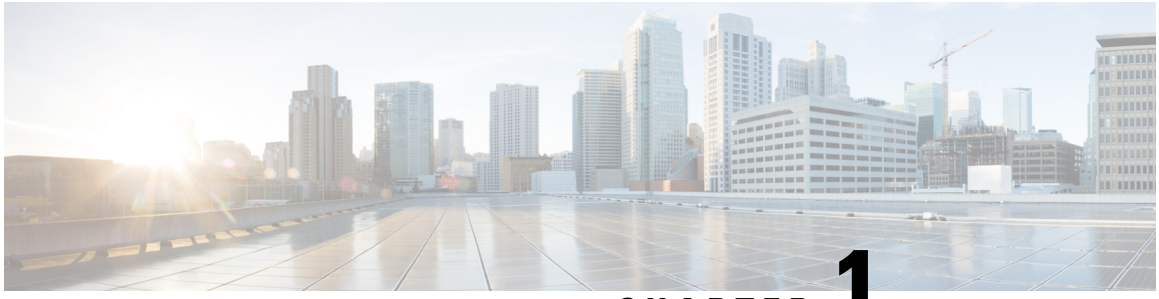
Stack Manager and High Availability Configuration Guide, Cisco IOS XE Fuji 16.9.x (Catalyst 3850 Switches)

First Published: 2018-07-18

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Managing Switch Stacks

Prerequisites for Switch Stacks

All the switches in the switch stack need to be running the same license level as the active switch. For information about license levels, see the *System Management Configuration Guide (Catalyst 3850 Switches)*.

All switches in the switch stack need to be running compatible software versions.

Restrictions for Switch Stacks

The following are restrictions for your switch stack configuration:

- Switch stacks running the LAN Base license level do not support Layer 3 features.
- A switch stack can have up to nine stacking-capable switches connected through their StackWise-480 ports.
- You cannot have a switch stack containing a mix of Catalyst 3850 and Catalyst 3650 switches.
- You cannot have a switch stack containing a mix of different license levels.

Note In a mixed stack configuration, there is limited support for some features. For more information about a specific feature, see the relevant Catalyst 3850 configuration guide.

Information About Switch Stacks

Switch Stack Overview

A switch stack can have up to nine stacking-capable switches connected through their StackWise-480 ports. The stack members work together as a unified system. Layer 2 and Layer 3 protocols present the entire switch stack as a single entity to the network.

A switch stack always has one active switch and one standby switch. If the active switch becomes unavailable, the standby switch assumes the role of the active switch, and continues to the keep the stack operational.

The active switch controls the operation of the switch stack, and is the single point of stack-wide management. From the active switch, you configure:

- System-level (global) features that apply to all stack members
- Interface-level features for each stack member

The active switch contains the saved and running configuration files for the switch stack. The configuration files include the system-level settings for the switch stack and the interface-level settings for each stack member. Each stack member has a current copy of these files for back-up purposes.

A switch stack can have up to nine stacking-capable switches connected through their StackWise-480 ports. The switches in the stack must be homogeneous and must have the same license level.

Supported Features in a Switch Stack

The system-level features supported on the active switch are supported on the entire switch stack.

Encryption Features

If the active switch is running the cryptographic universal software image (supports encryption), the encryption features are available on the switch stack.

StackWise-480

The stack members use the StackWise-480 technology to work together as a unified system. Layer 2 and Layer 3 protocols support the entire switch stack as a single entity in the network.

Note Switch stacks running the LAN Base image do not support Layer 3 features.

StackWise-480 has a stack bandwidth of 480 Gbps, and uses stateful switchover (SSO) to provide resiliency within the stack. The stack behaves as a single switching unit that is managed by an active switch elected by the member switches. The active switch automatically elects a standby switch within the stack. The active switch creates and updates all the switching, routing and wireless information and constantly synchronizes that information with the standby switch. If the active switch fails, the standby switch assumes the role of the active switch and continues to the keep the stack operational. Access points continue to remain connected during an active-to-standby switchover unless the access point is directly connected to the active switch. In this case the access point will lose power and reboot. A working stack can accept new members or delete old ones without service interruption.

Fast Stack Convergence

When a single link in a full ring stack becomes inoperable, there is a disruption in the forwarding of packets, and the stack moves to a half ring. With Catalyst 3850 switches this disruption of traffic (or stack convergence time) takes milliseconds.

StackPower

StackPower allows the power supplies in a stack to be shared as a common resource among all the switches in the stack. StackPower unifies the individual power supplies installed in the switches and creates a pool of power, directing that power where it is needed. Up to four switches can be configured in a StackPower stack using the StackPower cable.

For more information about StackPower, see the *Interface and Hardware Component Configuration Guide (Catalyst 3850 Switches)* .

Switch Stack Membership

A switch stack has up to stack members connected through their stack ports. A switch stack always has one active switch.

A standalone device is a device stack with one stack member that also operates as the active switch. You can connect one standalone device to another to create a stack containing two stack members, with one of them as the active switch. You can connect standalone devices to an existing device stack to increase the stack membership.

Hello messages are sent and received by all stack members.

- If a stack member does not respond, that member is removed from the stack.
- If the standby device does not respond, a new standby device is elected.
- If the active device does not respond, the standby device becomes the active device.

In addition, keepalive messages are sent and received between the active and standby devices.

- If the standby device does not respond, a new standby device is elected.
- If the active device does not respond, the standby device becomes the active device.

Changes to Switch Stack Membership

If you replace a stack member with an identical model, the new switch functions with exactly the same configuration as the replaced switch, assuming that the new switch (referred to as the provisioned switch) is using the same member number as the replaced switch.

The operation of the switch stack continues uninterrupted during membership changes unless you remove the active switch or you add powered-on standalone switches or switch stacks.

- Adding powered-on switches (merging) causes all switches to reload and elect a new active switch from among themselves. The newly elected active switch retains its role and configuration. All other switches retain their stack member numbers and use the stack configuration of the newly elected active switch.

Note In Cisco IOS XE 3.6.4E and later versions, when a new switch is powered-on as a standalone switch before it is added as part of the switch stack, only this switch is reloaded and not the whole switch stack.

- Removing powered-on stack members causes the switch stack to divide (partition) into two or more switch stacks, each with the same configuration. This can cause:
  - An IP address conflict in your network. If you want the switch stacks to remain separate, change the IP address or addresses of the newly created switch stacks.
  - A MAC address conflict between two members in the stack. You can use the **stack-mac update force** command to resolve the conflict.

If a newly created switch stack does not have an active switch or standby switch, the switch stack will reload and elect a new active switch.

Note Make sure that you power off the switches that you add to or remove from the switch stack.

After adding or removing stack members, make sure that the switch stack is operating at full bandwidth (480 Gbps). Press the Mode button on a stack member until the Stack mode LED is on. The last two right port LEDs on all switches in the stack should be green. Depending on the switch model, the last two right ports are 10-Gigabit Ethernet ports or small form-factor pluggable (SFP) module ports (10/100/1000 ports). If one or both of these LEDs are not green on any of the switches, the stack is not operating at full bandwidth.

If you remove powered-on members but do not want to partition the stack:

- Power off the switches in the newly created switch stacks.
- Reconnect them to the original switch stack through their stack ports.
- Power on the switches.

For cabling and power considerations that affect switch stacks, see the *Catalyst 3850 Switch Hardware Installation Guide* .

Stack Member Numbers

The stack member number (1 to 9) identifies each member in the switch stack. The member number also determines the interface-level configuration that a stack member uses. You can display the stack member number by using the **show switch** EXEC command.

A new, out-of-the-box device (one that has not joined a device stack or has not been manually assigned a stack member number) ships with a default stack member number of 1. When it joins a device stack, its default stack member number changes to the lowest available member number in the stack.

Stack members in the same stack cannot have the same stack member number. Every stack member, including a standalone device, retains its member number until you manually change the number or unless the number is already being used by another member in the stack.

- If you manually change the stack member number by using the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command, the new number goes into effect after that stack member resets (or after you use the **reload slot** *stack-member-number* privileged EXEC command) and only if that number is not already assigned to any other members in the stack. Another way to change the stack member number is by changing the device_NUMBER environment variable.

  If the number is being used by another member in the stack, the device selects the lowest available number in the stack.

  If you manually change the number of a stack member and no interface-level configuration is associated with that new member number, that stack member resets to its default configuration.

  You cannot use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command on a provisioned device. If you do, the command is rejected.

- If you move a stack member to a different device stack, the stack member retains its number only if the number is not being used by another member in the stack. If it is being used, the device selects the lowest available number in the stack.

- If you merge device stacks, the device that join the device stack of a new active device select the lowest available numbers in the stack.

As described in the hardware installation guide, you can use the device port LEDs in Stack mode to visually determine the stack member number of each stack member.

In the **default** mode Stack LED will blink in green color only on the active switch. However, when we scroll the Mode button to **Stack** option - Stack LED will glow green on all the stack members.

When mode button is scrolled to **Stack** option, the switch number of each stack member will be displayed as LEDs on the first five ports of that switch. The switch number is displayed in binary format for all stack members. On the switch, the amber LED indicates value 0 and green LED indicates value 1.

Example for switch number 5 (Binary - 00101):

First five LEDs glow as follows on stack member with switch number 5.

- Port-1 : Amber
- Port-2 : Amber
- Port-3 : Green
- Port-4 : Amber
- Port-5 : Green

Similarly, the first five LEDs glow amber or green, depending on the switch number on all stack members.

Note

- If you connect a Horizontal stack port to a normal network port on other end, stack port transmission/reception will be disabled within 30 seconds if no SDP packets are received from the other end.
- Stack port will not go down but only transmission/reception will be disabled. The log message shown below will be displayed on the console. Once the peer end network port is converted to stack port, transmission/reception on this stack port will be enabled.

```
%STACKMGR-4-HSTACK_LINK_CONFIG: Verify peer stack port setting for hstack
 StackPort-1 switch 5 (hostname-switchnumber)
```

Stack Member Priority Values

A higher priority value for a stack member increases the probability of it being elected active switch and retaining its stack member number. The priority value can be 1 to 15. The default priority value is 1. You can display the stack member priority value by using the **show switch** EXEC command.

Note We recommend assigning the highest priority value to the device that you prefer to be the active device. This ensures that the device is reelected as the active device if a reelection occurs.

To change the priority value for a stack member, use the **switch** *stack-member-number* **priority** *new priority-value* EXEC command. For more information, see the "Setting the Stack Member Priority Value" section.

The new priority value takes effect immediately but does not affect the current active device. The new priority value helps determine which stack member is elected as the new active device when the current active device or the device stack resets.

Switch Stack Bridge ID and MAC Address

A switch stack is identified in the network by its *bridge ID* and, if it is operating as a Layer 3 device, its router MAC address. The bridge ID and router MAC address are determined by the MAC address of the active switch.

If the active switch changes, the MAC address of the new active switch determines the new bridge ID and router MAC address.

If the entire switch stack reloads, the switch stack uses the MAC address of the active switch.

Persistent MAC Address on the Switch Stack

You can use the persistent MAC address feature to set a time delay before the stack MAC address changes. During this time period, if the previous active switch rejoins the stack, the stack continues to use its MAC address as the stack MAC address, even if the switch is now a stack member and not an active switch. If the previous active switch does not rejoin the stack during this period, the switch stack takes the MAC address of the new active switch as the stack MAC address. By default, the stack MAC address will be the MAC address of the first active switch, even if a new active switch takes over.

You can also configure stack MAC persistency so that the stack MAC address never changes to the new active switch MAC address.

Active and Standby Switch Election and Reelection

All stack members are eligible to be the active switch or the standby switch. If the active switch becomes unavailable, the standby switch becomes the active switch.

An active switch retains its role unless one of these events occurs:

- The switch stack is reset.
- The active switch is removed from the switch stack.
- The active switch is reset or powered off.
- The active switch fails.
- The switch stack membership is increased by adding powered-on standalone switches or switch stacks.

The active switch is elected or reelected based on one of these factors and in the order listed:

1. The switch that is currently the active switch.
2. The switch with the highest stack member priority value.

Note We recommend assigning the highest priority value to the switch that you prefer to be the active switch. This ensures that the switch is reelected as active switch if a reelection occurs.

3. The switch with the shortest start-up time. Differences in start-up times between the feature image licenses determine the active switch. For example, a switch running the IP Services license level has a higher priority than the switch running the IP Base license level, but the switch running the IP Base license level becomes the active switch because the other switch takes 120 seconds longer to start. To avoid this problem, upgrade the switch running the IP Base license level to the same licensed feature set and software image as the other switch, or manually start the active switch and wait at least 8 seconds before starting the new member switch that is running the IP Base license level.
4. The switch with the lowest MAC address.

Note The factors for electing or reelecting a new standby switch are same as those for the active switch election or reelection, and are applied to all participating switches except the active switch.

After election, the new active switch becomes available after a few seconds. In the meantime, the switch stack uses the forwarding tables in memory to minimize network disruption. The physical interfaces on the other available stack members are not affected during a new active switch election and reset.

When the previous active switch becomes available, it *does not* resume its role as the active switch.

If you power on or reset an entire switch stack, some stack members *might not* participate in the active switch election. Stack members that are powered on within the same 2-minute timeframe participate in the active switch election and have a chance to become the active switch. Stack members that are powered on after the

120-second timeframe do not participate in this initial election and become stack members. For powering considerations that affect active-switch elections, see the switch hardware installation guide.

As described in the hardware installation guide, you can use the ACTV LED on the switch to see if the switch is the active switch.

Switch Stack Configuration Files

The active switch has the saved and running configuration file for the switch stack. The standby switch automatically receives the synchronized running configuration file. Stack members receive synchronized copies when the running configuration file is saved into the startup configuration file. If the active switch becomes unavailable, the standby switch takes over with the current running configuration.

The configuration files record these settings:

- System-level (global) configuration settings such as IP, STP, VLAN, and SNMP settings that apply to all stack members
- Stack member interface-specific configuration settings that are specific for each stack member

Note The interface-specific settings of the active switch are saved if the active switch is replaced without saving the running configuration to the startup configuration.

A new, out-of-box device joining a switch stack uses the system-level settings of that switch stack. If a device is moved to a different switch stack before it is powered on, that device loses its saved configuration file and uses the system-level configuration of the new switch stack. If the device is powered on as a standalone device before it joins the new switch stack, the stack will reload. When the stack reloads, the new device may become the device, retain its configuration and overwrite the configuration files of the other stack members.

The interface-specific configuration of each stack member is associated with the stack member number. Stack members retain their numbers unless they are manually changed or they are already used by another member in the same switch stack. If the stack member number changes, the new number goes into effect after that stack member resets.

- If an interface-specific configuration does not exist for that member number, the stack member uses its default interface-specific configuration.
- If an interface-specific configuration exists for that member number, the stack member uses the interface-specific configuration associated with that member number.

If you replace a failed member with an identical model, the replacement member automatically uses the same interface-specific configuration as the failed device. You do not need to reconfigure the interface settings. The replacement device (referred to as the provisioned device) must have the same stack member number as the failed device.

You back up and restore the stack configuration in the same way as you would for a standalone device configuration.

Offline Configuration to Provision a Stack Member

You can use the offline configuration feature to *provision* (to supply a configuration to) a new switch before it joins the switch stack. You can configure the stack member number, the switch type, and the interfaces associated with a switch that is not currently part of the stack. The configuration that you create on the switch stack is called the *provisioned configuration*. The switch that is added to the switch stack and that receives this configuration is called the *provisioned switch*.

You manually create the provisioned configuration through the **switch** *stack-member-number* **provision** *type* global configuration command. You must change the *stack-member-number* on the provisioned switch before you add it to the stack, and it must match the stack member number that you created for the new switch on the switch stack. The switch type in the provisioned configuration must match the switch type of the newly added switch. The provisioned configuration is automatically created when a switch is added to a switch stack and when no provisioned configuration exists.

When you configure the interfaces associated with a provisioned switch, the switch stack accepts the configuration, and the information appears in the running configuration. However, as the switch is not active, any configuration on the interface is not operational and the interface associated with the provisioned switch does not appear in the display of the specific feature. For example, VLAN configuration information associated with a provisioned switch does not appear in the **show vlan** user EXEC command output on the switch stack.

The switch stack retains the provisioned configuration in the running configuration whether or not the provisioned switch is part of the stack. You can save the provisioned configuration to the startup configuration file by entering the **copy running-config startup-config** privileged EXEC command. The startup configuration file ensures that the switch stack can reload and can use the saved information whether or not the provisioned switch is part of the switch stack.

Effects of Adding a Provisioned Switch to a Switch Stack

When you add a provisioned Device to the switch stack, the stack applies either the provisioned configuration or the default configuration. This table lists the events that occur when the switch stack compares the provisioned configuration with the provisioned switch.

Table 1: Results of Comparing the Provisioned Configuration with the Provisioned Switch

Scenario		Result
The stack member numbers and the Device types match.	**1.** If the stack member number of the provisioned switch matches the stack member number in the provisioned configuration on the stack, and **2.** If the Device type of the provisioned switch matches the Device type in the provisioned configuration on the stack.	The switch stack applies the provisioned configuration to the provisioned switch and adds it to the stack.

Scenario		Result
The stack member numbers match but the Device types do not match.	**1.** If the stack member number of the provisioned switch matches the stack member number in the provisioned configuration on the stack, but **2.** The Device type of the provisioned switch does not match the Device type in the provisioned configuration on the stack.	The switch stack applies the default configuration to the provisioned switch and adds it to the stack. The provisioned configuration is changed to reflect the new information.
The stack member number is not found in the provisioned configuration.		The switch stack applies the default configuration to the provisioned switch and adds it to the stack. The provisioned configuration is changed to reflect the new information.
The stack member number of the provisioned switch is not found in the provisioned configuration.		The switch stack applies the default configuration to the provisioned switch and adds it to the stack.

If you add a provisioned switch that is a different type than specified in the provisioned configuration to a powered-down switch stack and then apply power, the switch stack rejects the (now incorrect) **switch** *stack-member-number* **provision** *type* global configuration command in the startup configuration file. However, during stack initialization, the nondefault interface configuration information in the startup configuration file for the provisioned interfaces (potentially of the wrong type) is executed. Depending on the differences between the actual Device type and the previously provisioned switch type, some commands are rejected, and some commands are accepted.

Note If the switch stack does not contain a provisioned configuration for a new Device, the Device joins the stack with the default interface configuration. The switch stack then adds to its running configuration with a **switch** *stack-member-number* **provision** *type* global configuration command that matches the new Device. For configuration information, see the *Provisioning a New Member for a Switch Stack* section.

Effects of Replacing a Provisioned Switch in a Switch Stack

When a provisioned switch in a switch stack fails, it is removed from the stack, and is replaced with another Device, the stack applies either the provisioned configuration or the default configuration to it. The events that occur when the switch stack compares the provisioned configuration with the provisioned switch are the same as those when you add a provisioned switch to a stack.

Effects of Removing a Provisioned Switch from a Switch Stack

If you remove a provisioned switch from the switch stack, the configuration associated with the removed stack member remains in the running configuration as provisioned information. To completely remove the configuration, use the **no switch** *stack-member-number* **provision** global configuration command.

Upgrading a Switch Running Incompatible Software

The auto-upgrade and auto-advise features enable a switch with software packages that are incompatible with the switch stack to be upgraded to a compatible software version so that it can join the switch stack.

Auto-Upgrade

The purpose of the auto-upgrade feature is to allow a switch to be upgraded to a compatible software image, so that the switch can join the switch stack.

The switch with the higher version of software is made the active switch and all other switches that are to be upgraded are booted simultaneously. If you have new switches to add to the stack, first power them off, add them to the stack and then boot them simultaneously. You cannot add more members to a stack when an auto-upgrade is going on in the stack. You can add new members only after the on-going auto-upgrade process is completed.

When a new switch attempts to join a switch stack, each stack member performs compatibility checks with itself and the new switch. Each stack member sends the results of the compatibility checks to the active stack, which uses the results to determine whether the switch can join the switch stack. If the software on the new switch is incompatible with the switch stack, the new switch enters version-mismatch (VM) mode.

If the auto-upgrade feature is enabled on the existing switch stack, the active stack automatically upgrades the new switch with the same software image running on a compatible stack member. Auto-upgrade starts a few minutes after the mismatched software is detected before starting.

You can perform auto-upgrade on the newly added member of a stack only after the existing members of the stack are already auto-upgraded.

Auto-upgrade is disabled by default.

Note the following limitations before starting an auto-upgrade:

- Do not perform an auto-upgrade in bundle mode.
- Do not perform an auto-upgrade in half-ring stack.
- Do not perform stack merge of two active switches that have different version of images.
- Do not perform staggered boot of the switches to be upgraded.

Auto-upgrade includes an auto-copy process and an auto-extract process.

- Auto-copy automatically copies the software image running on any stack member to the new switch to automatically upgrade it. Auto-copy occurs if auto-upgrade is enabled, if there is enough flash memory in the new switch, and if the software image running on the switch stack is suitable for the new switch.

  Note A switch in VM mode might not run all released software. For example, new switch hardware is not recognized in earlier versions of software.

- Automatic extraction (auto-extract) occurs when the auto-upgrade process cannot find the appropriate software in the stack to copy to the new switch. In that case, the auto-extract process searches all switches in the stack for the bin file needed to upgrade the switch stack or the new switch. The bin file can be in any flash file system in the switch stack or in the new switch. If a bin file suitable for the new switch is found on a stack member, the process extracts the file and automatically upgrades the new switch.

The auto-upgrade feature is not available in bundle mode. The switch stack must be running in installed mode. If the switch stack is in bundle mode, use the **software expand** privileged EXEC command to change to installed mode.

You can enable auto-upgrade by using the **software auto-upgrade enable** global configuration command on the new switch. You can check the status of auto-upgrade by using the **show running-config** privileged EXEC command and by checking the *Auto upgrade* line in the display.

You can configure auto-upgrade to upgrade the new switch with a specific software bundle by using the **software auto-upgrade source url** global configuration command. If the software bundle is invalid, the new switch is upgraded with the same software image running on a compatible stack member.

When the auto-upgrade process is complete, the new switch reloads and joins the stack as a fully functioning member. If you have both stack cables connected during the reload, network downtime does not occur because the switch stack operates on two rings.

For more information about upgrading a switch running incompatible software see the *Cisco IOS File System, Configuration Files, and Bundle Files Appendix, Cisco IOS XE Release 3SE (Catalyst 3850 Switches)*.

Auto-Advise

The auto-advise feature is triggered when:

- The auto-upgrade feature is disabled.
- The new switch is in bundle mode and the stack is in installed mode. Auto-advise displays syslog messages about using the **software auto-upgrade** privileged EXEC command to change the new switch to installed mode.
- The stack is in bundle mode. Auto-advise displays syslog messages about booting the new switch in bundle mode so that it can join the stack.
- An auto-upgrade attempt fails because the new switch is running incompatible software. After the switch stack performs compatibility checks with the new switch, auto-advise displays syslog messages about whether the new switch can be auto-upgraded.

Auto-advise cannot be disabled. It does *not* give suggestions when the switch stack software and the software of the switch in version-mismatch (VM) mode do not contain the same license level.

Examples of Auto-Advise Messages

Auto-Upgrade Is Disabled and Incompatible Switch Attempting to Join: Example

This sample auto-advise output shows the system messages displayed when the auto-upgrade feature is disabled and an incompatible switch 1 tries to join the switch stack:

```
*Oct 18 08:36:19.379: %INSTALLER-6-AUTO_ADVISE_SW_INITIATED: 2 installer:  Auto advise
initiated for switch 1
*Oct 18 08:36:19.380: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  Searching stack for software
 to upgrade switch 1
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  Switch 1 with incompatible
 software has been
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  added to the stack. The
software running on
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  all stack members was
scanned and it has been
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  determined that the 'software
 auto-upgrade'
```

```
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  command can be used to
install compatible
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  software on switch 1.
```

Auto-Upgrade is Disabled and New Switch is in Bundle Mode: Example

This sample auto-advise output shows the system messages displayed when auto-upgrade is disabled and a switch running in bundle mode tries to join the stack that is running in installed mode:

```
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW_INITIATED: 2 installer:  Auto advise
initiated for switch 1
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  Switch 1 running bundled
software has been added
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  to the stack that is running
 installed software.
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  The 'software auto-upgrade'
 command can be used to
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  convert switch 1 to the
installed running mode by
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer:  installing its running
software.
```

Switch Stack Management Connectivity

You manage the switch stack and the stack member interfaces through the active switch. You can use the CLI, SNMP, and supported network management applications such as CiscoWorks. You cannot manage stack members on an individual device basis.

Note Use SNMP to manage network features across the stack that are defined by supported MIBs. The switch does not support MIBs to manage stacking-specific features such as stack membership and election.

Connectivity to the Switch Stack Through an IP Address

The switch stack is managed through a single IP address. The IP address is a system-level setting and is not specific to the active stack or to any other stack member. You can still manage the stack through the same IP address even if you remove the active stack or any other stack member from the stack, provided there is IP connectivity.

Note Stack members retain their IP addresses when you remove them from a switch stack. To avoid a conflict by having two devices with the same IP address in your network, change the IP addresses of any active stack that you remove from the switch stack.

For related information about switch stack configurations, see the *Switch Stack Configuration Files section*.

Connectivity to the Switch Stack Through Console Ports or Ethernet Management Ports

You can connect to the active switch by using one of these methods:

- You can connect a terminal or a PC to the active switch through the console port of one or more stack members.
- You can connect a PC to the active switch through the Ethernet management ports of one or more stack members. For more information about connecting to the switch stack through Ethernet management ports, see the *Using the Ethernet Management Port section*.

You can connect to the active switch by connecting a terminal or a PC to the active switch through the console port of one or more stack members.

When you use the console port of a stack member, a VTY session is created with the IP address in the 192.168.0.1/24 subnet.

Be careful when using multiple CLI sessions to the active switch. Commands that you enter in one session are not displayed in the other sessions. Therefore, it is possible that you might not be able to identify the session from which you entered a command.

We recommend using only one CLI session when managing the switch stack.

How to Configure a Switch Stack

Enabling the Persistent MAC Address Feature

Note When you enter the command to configure this feature, a warning message appears with the consequences of your configuration. You should use this feature cautiously. Using the old active switch MAC address elsewhere in the same domain could result in lost traffic.

Follow these steps to enable persistent MAC address:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **stack-mac persistent timer** [**0** | *time-value*]
4. **end**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**stack-mac persistent timer** [**0** | *time-value*] **Example:** `Device(config)# stack-mac persistent timer 7`	Enables a time delay after a stack-active switch change before the stack MAC address changes to that of the new ac. If the previous active switch rejoins the stack during this period, the stack uses that MAC address as the stack MAC address. You can configure the time period as 0 to 60 minutes. • Enter the command with no value to set the default delay of approximately 4 minutes. We recommend that you always enter a value. If the command is entered without a value, the time delay appears in the running-config file with an explicit timer value of 4 minutes. • Enter **0** to continue using the MAC address of the current active switch indefinitely. The stack MAC address of the previous active switch is used until you enter the **no stack-mac persistent timer** command, which immediately changes the stack MAC address to that of the current active switch. • Enter a *time-value* from 1 to 60 minutes to configure the time period before the stack MAC address changes to the new active switch. The stack MAC address of the previous active switch is used until the configured time period expires or until you enter the **no stack-mac persistent timer** command. **Note** If you enter the **no stack-mac persistent timer** command after a new active switch takes over, before the time expires, the switch stack moves to the current active switch MAC address.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

What to do next

Use the **no stack-mac persistent timer** global configuration command to disable the persistent MAC address feature.

Assigning a Stack Member Number

This optional task is available only from the active stack.

Follow these steps to assign a member number to a stack member:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **switch** *current-stack-member-number* **renumber** *new-stack-member-number*
4. **end**
5. **reload slot** *stack-member-number*
6. **show switch**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**switch** *current-stack-member-number* **renumber** *new-stack-member-number* **Example:** `Device# switch 3 renumber 4`	Specifies the current stack member number and the new stack member number for the stack member. The range is 1 to 9. You can display the current stack member number by using the **show switch** user EXEC command.
Step 4	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 5	**reload slot** *stack-member-number* **Example:** `Device# reload slot 4`	Resets the stack member.
Step 6	**show switch** **Example:** `showDevice`	Verify the stack member number.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Setting the Stack Member Priority Value

This optional task is available only from the active stack.

Follow these steps to assign a priority value to a stack member:

SUMMARY STEPS

1. **enable**
2. **switch** *stack-member-number* **priority** *new-priority-number*
3. **show switch** *stack-member-number*
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**switch** *stack-member-number* **priority** *new-priority-number* **Example:** `Device# switch 3 priority 2`	Specifies the stack member number and the new priority for the stack member. The stack member number range is 1 to 9. The priority value range is 1 to 15. You can display the current priority value by using the **show switch** user EXEC command. The new priority value takes effect immediately but does not affect the current active stack. The new priority value

	Command or Action	Purpose
		helps determine which stack member is elected as the new active stack when the current active stack or switch stack resets.
Step 3	**show switch** *stack-member-number* **Example:** `Device# show switch`	Verify the stack member priority value.
Step 4	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Provisioning a New Member for a Switch Stack

This optional task is available only from the active switch.

SUMMARY STEPS

1. **show switch**
2. **configure terminal**
3. **switch** *stack-member-number* **provision** *type*
4. **end**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show switch** **Example:** `Device# show switch`	Displays summary information about the switch stack.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**switch** *stack-member-number* **provision** *type* **Example:** `Device(config)# switch 3 provision WS-xxxx`	Specifies the stack member number for the preconfigured switch. By default, no switches are provisioned. For *stack-member-number*, the range is 1 to 9. Specify a stack member number that is not already used in the switch stack. See Step 1. For *type*, enter the model number of a supported switch that is listed in the command-line help strings.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Removing Provisioned Switch Information

Before you begin, you must remove the provisioned switch from the stack. This optional task is available only from the active stack.

SUMMARY STEPS

1. **configure terminal**
2. **no switch** *stack-member-number* **provision**
3. **end**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**no switch** *stack-member-number* **provision** **Example:** `Device(config)# no switch 3 provision`	Removes the provisioning information for the specified member.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Example

If you are removing a provisioned switch in a stack with this configuration:

- The stack has four members
- Stack member 1 is the active stack
- Stack member 3 is a provisioned switch

and want to remove the provisioned information and to avoid receiving an error message, you can remove power from stack member 3, disconnect the StackWise-480 cables between the stack member 3 and switches to which it is connected, reconnect the cables between the remaining stack members, and enter the **no switch** *stack-member-number* **provision** global configuration command.

Displaying Incompatible Switches in the Switch Stack

SUMMARY STEPS

1. **show switch**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show switch** **Example:** `Device# show switch`	Displays any incompatible switches in the switch stack (indicated by a 'Current State' of 'V-Mismatch'). The V-Mismatch state identifies the switches with incompatible software. The output displays Lic-Mismatch for switches that are not running the same license level as the active switch. For information about managing license levels, see the *System Management Configuration Guide (Catalyst 3850 Switches)*.

Upgrading an Incompatible Switch in the Switch Stack

Before you begin

- Ensure the switches are install booted.
- Ensure that the stack is connected in full ring mode.

SUMMARY STEPS

1. **software auto-upgrade**
2. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**software auto-upgrade** **Example:** `Device# software auto-upgrade`	Upgrades incompatible switches in the switch stack, or changes switches in bundle mode to installed mode.
Step 2	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Troubleshooting the Switch Stack

Temporarily Disabling a Stack Port

If a stack port is flapping and causing instability in the stack ring, to disable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **disable** privileged EXEC command. To reenable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **enable** command.

Note Be careful when using the **switch** *stack-member-number* **stack port** *port-number* **disable** command. When you disable the stack port, the stack operates at half bandwidth.

A stack is in the full-ring state when all members are connected through the stack ports and are in the ready state.

The stack is in the partial-ring state when the following occurs:

- All members are connected through their stack ports but some are not in the ready state.
- Some members are not connected through the stack ports.

SUMMARY STEPS

1. **switch** *stack-member-number* **stack port** *port-number* **disable**
2. **switch** *stack-member-number* **stack port** *port-number* **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**switch** *stack-member-number* **stack port** *port-number* **disable** **Example:**	Disables the specified stack port.

	Command or Action	Purpose
	`Device# switch 2 stack port 1 disable`	
Step 2	**switch** *stack-member-number* **stack port** *port-number* **enable** **Example:** `Device# switch 2 stack port 1 enable`	Reenables the stack port.

When you disable a stack port and the stack is in the full-ring state, you can disable only one stack port. This message appears:

```
Enabling/disabling a stack port may cause undesired stack changes. Continue?[confirm]
```

When you disable a stack port and the stack is in the partial-ring state, you cannot disable the port. This message appears:

```
Disabling stack port not allowed with current stack configuration.
```

Reenabling a Stack Port While Another Member Starts

Stack Port 1 on Switch 1 is connected to Port 2 on Switch 4. If Port 1 is flapping, you can disable Port 1 with the **switch 1 stack port 1 disable** privileged EXEC command. While Port 1 on Switch 1 is disabled and Switch 1 is still powered on, follow these steps to reenable a stack port:

Step 1 Disconnect the stack cable between Port 1 on Switch 1 and Port 2 on Switch 4.

Step 2 Remove Switch 4 from the stack.

Step 3 Add a switch to replace Switch 4 and assign it switch-number 4.

Step 4 Reconnect the cable between Port 1 on Switch 1 and Port 2 on Switch 4 (the replacement switch).

Step 5 Reenable the link between the switches. Enter the **switch 1 stack port 1 enable** privileged EXEC command to enable Port 1 on Switch 1.

Step 6 Power on Switch 4.

Caution Powering on Switch 4 before enabling the Port 1 on Switch 1 might cause one of the switches to reload.

If Switch 4 is powered on first, you might need to enter the **switch 1 stack port 1 enable** and the **switch 4 stack port 2 enable** privileged EXEC commands to bring up the link.

Monitoring the Device Stack

Table 2: Commands for Displaying Stack Information

Command	Description
show switch	Displays summary information about the stack, including the status of provisioned switches and switches in version-mismatch mode.
show switch *stack-member-number*	Displays information about a specific member.
show switch detail	Displays detailed information about the stack.
show switch neighbors	Displays the stack neighbors.
show switch stack-ports [**summary**]	Displays port information for the stack. Use the **summary** keyword to display the stack cable length, the stack link status, and the loopback status.
show redundancy	Displays the redundant system and the current processor information. The redundant system information includes the system uptime, standby failures, switchover reason, hardware, configured and operating redundancy mode. The current processor information displayed includes the active location, the software state, the uptime in the current state and so on.
show redundancy state	Displays all the redundancy states of the active and standby devices.

Configuration Examples for Switch Stacks

Switch Stack Configuration Scenarios

Most of these switch stack configuration scenarios assume that at least two device are connected through their ports.

Table 3: Configuration Scenarios

Scenario		Result
Active switch election specifically determined by existing active switches	Connect two powered-on switch stacks through the StackWise-480 ports.	Only one of the two active switches becomes the new active switch.

Scenario		Result
Active switch election specifically determined by the stack member priority value	**1.** Connect two switches through their ports. **2.** Use the **switch** *stack-member-number* **priority** *new-priority-number* global configurationEXEC command to set one stack member with a higher member priority value. **3.** Restart both stack members at the same time.	The stack member with the higher priority value is elected active switch.
Active switch election specifically determined by the configuration file	Assuming that both stack members have the same priority value: **1.** Make sure that one stack member has a default configuration and that the other stack member has a saved (nondefault) configuration file. **2.** Restart both stack members at the same time.	The stack member with the saved configuration file is elected active switch.
Active switch election specifically determined by the MAC address	Assuming that both stack members have the same priority value, configuration file, and feature setlicense level, restart both stack members at the same time.	The stack member with the lower MAC address is elected active switch.
Stack member number conflict	Assuming that one stack member has a higher priority value than the other stack member: **1.** Ensure that both stack members have the same stack member number. If necessary, use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* global configurationEXEC command. **2.** Restart both stack members at the same time.	The stack member with the higher priority value retains its stack member number. The other stack member has a new stack member number.
Add a stack member	**1.** Power off the new switch. **2.** Through their ports, connect the new switch to a powered-on switch stack. **3.** Power on the new switch.	The active switch is retained. The new switch is added to the switch stack.
Active switch failure	Remove (or power off) the active switch.	The standby switch becomes the new active switch. All other stack members in the stack remain as stack members and do not reboot.

Scenario		Result
Add more than nine stack members	**1.** Through their StackWise-480 ports, connect ten device. **2.** Power on all device.	Two device become active switches. One active switch has nine stack members. The other active switch remains as a standalone device. Use the Mode button and port LEDs on the device to identify which device are active switches and which device belong to each active switch.

Enabling the Persistent MAC Address Feature: Example

This example shows how to configure the persistent MAC address feature for a 7-minute time delay and to verify the configuration:

```
Device(config)# stack-mac persistent timer 7
  WARNING: The stack continues to use the base MAC of the old Master
  WARNING: as the stack MAC after a master switchover until the MAC
  WARNING: persistency timer expires. During this time the Network
  WARNING: Administrators must make sure that the old stack-mac does
  WARNING: not appear elsewhere in this network domain. If it does,
  WARNING: user traffic may be blackholed.
  Device(config)# end
  Device# show switch
  Switch/Stack Mac Address : 0016.4727.a900
  Mac persistency wait time: 7 mins
                                                H/W   Current
  Switch#  Role   Mac Address     Priority Version  State
  -------------------------------------------------------------
  *1        0016.4727.a900     1      P2B     Ready
```

Provisioning a New Member for a Switch Stack: Example

This example shows how to provision a switch with a stack member number of 2 for the switch stack. The **show running-config** command output shows the interfaces associated with the provisioned switch:

```
Device(config)# switch 2 provision switch_PID
  Device(config)# end
  Device# show running-config | include switch 2
  !
  interface GigabitEthernet2/0/1
  !
  interface GigabitEthernet2/0/2
  !
  interface GigabitEthernet2/0/3
  <output truncated>
```

show switch stack-ports summary Command Output: Example

Only Port 1 on stack member 2 is disabled.

```
Device# show switch stack-ports summary
  Device#/  Stack   Neighbor   Cable    Link   Link   Sync      #         In
   Port#     Port              Length    OK   Active   OK    Changes   Loopback
            Status                                          To LinkOK
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     OK         3      50 cm     Yes    Yes    Yes        1        No
    1/2     Down      None    3 m       Yes    No     Yes        1        No
    2/1     Down      None    3 m       Yes    No     Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     OK         1      50 cm     Yes    Yes    Yes        1        No
```

Table 4: show switch stack-ports summary Command Output

Field	Description
Switch#/Port#	Member number and its stack port number.
Stack Port Status	Status of the stack port. • Absent—No cable is detected on the stack port. • Down—A cable is detected, but either no connected neighbor is up, or the stack port is disabled. • OK—A cable is detected, and the connected neighbor is up.
Neighbor	Switch number of the active member at the other end of the stack cable.
Cable Length	Valid lengths are 50 cm, 1 m, or 3 m. If the switch cannot detect the cable length, the value is *no cable*. The cable might not be connected, or the link might be unreliable.
Link OK	Whether the stack cable is connected and functional. There may or may not be a neighbor connected on the other end. The *link partner* is a stack port on a neighbor switch. • No—There is no stack cable connected to this port or the stack cable is not functional. • Yes—There is a functional stack cable connected to this port.
Link Active	Whether a neighbor is connected on the other end of the stack cable. • No—No neighbor is detected on the other end. The port cannot send traffic over this link. • Yes—A neighbor is detected on the other end. The port can send traffic over this link.
Sync OK	Whether the link partner sends valid protocol messages to the stack port. • No—The link partner does not send valid protocol messages to the stack port. • Yes—The link partner sends valid protocol messages to the port.

Field	Description
# Changes to LinkOK	The relative stability of the link. If a large number of changes occur in a short period of time, link flapping can occur.
In Loopback	Whether a stack cable is attached to a stack port on the member. • No—At least one stack port on the member has an attached stack cable. • Yes—None of the stack ports on the member has an attached stack cable.

Software Loopback: Examples

In a stack with three members, stack cables connect all the members:

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync   #Changes   In
            Status            Length    OK    Active  OK     To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------   --------
    1/1     OK         3      50 cm     Yes    Yes    Yes        1        No
    1/2     OK         2      3 m       Yes    Yes    Yes        1        No
    2/1     OK         1      3 m       Yes    Yes    Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     OK         1      50 cm     Yes    Yes    Yes        1        No
```

If you disconnect the stack cable from Port 1 on Switch 1, these messages appear:

```
01:09:55: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 3 has changed to state DOWN
  01:09:56: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 1 has changed to state DOWN

  Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback

  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable  No     No     No        1        No
    1/2     OK         2      3 m       Yes    Yes    Yes       1        No
    2/1     OK         1      3 m       Yes    Yes    Yes       1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes       1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes       1        No
    3/2     Down      None    50 cm     No     No     No        1        No
```

If you disconnect the stack cable from Port 2 on Switch 1, the stack splits.

Switch 2 and Switch 3 are now in a two-member stack connected through stack cables:

```
Device# show sw stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    2/1     Down      None    3 m       No     No     No        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes       1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes       1        No
```

```
   3/2     Down      None    50 cm     No     No     No         1        No
```

Switch 1 is a standalone switch:

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable  No     No     No         1        Yes
    1/2     Absent    None    No cable  No     No     No         1        Yes
```

Software Loopback with Connected Stack Cables: Examples

- On Port 1 on Switch 1, the port status is *Down*, and a cable is connected.

  On Port 2 on Switch 1, the port status is *Absent*, and no cable is connected.

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Down      None    50 Cm      No     No     No        1          No
    1/2     Absent    None    No cable   No     No     No        1          No
```

- In a *physical loopback*, a cable connects both stack ports on a switch. You can use this configuration to test
  - Cables on a switch that is running properly
  - Stack ports with a cable that works properly

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    2/1       OK       2      50 cm     Yes    Yes    Yes       1         No
    2/2       OK       2      50 cm     Yes    Yes    Yes       1         No
```

  The port status shows that
  - Switch 2 is a standalone switch.
  - The ports can send and receive traffic.

Software Loopback with no Connected Stack Cable: Example

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable   No     No     No        1         Yes
    1/2     Absent    None    No cable   No     No     No        1         Yes
```

Finding a Disconnected Stack Cable: Example

Stack cables connect all stack members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status for the members:

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     OK         2      50 cm     Yes    Yes    Yes        0        No
    1/2     OK         2      50 cm     Yes    Yes    Yes        0        No
    2/1     OK         1      50 cm     Yes    Yes    Yes        0        No
    2/2     OK         1      50 cm     Yes    Yes    Yes        0        No
```

If you disconnect the cable from Port 2 on Switch 1, these messages appear:

```
%STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 2 has changed to state DOWN

  %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 1 has changed to state DOWN
```

This is now the port status:

```
  Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1       OK       2      50 cm     Yes    Yes    Yes        1        No
    1/2     Absent    None    No cable   No     No     No        2        No
    2/1      Down     None    50 cm      No     No     No        2        No
    2/2       OK       1      50 cm     Yes    Yes    Yes        1        No
```

Only one end of the cable connects to a stack port, Port 1 on Switch 2.

- The *Stack Port Status* value for Port 2 on Switch 1 is *Absent*, and the value for Port 1 on Switch 2 is *Down*.
- The *Cable Length* value is *No cable*.

Diagnosing the problem:

- Verify the cable connection for Port 2 on Switch 1.
- Port 2 on Switch 1 has a port or cable problem if
  - The *In Loopback* value is *Yes*.

  or

  - The *Link OK*, *Link Active*, or *Sync OK* value is *No*.

Fixing a Bad Connection Between Stack Ports: Example

Stack cables connect all members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status:

```
Device# show switch stack-ports summary
  Device#
 Sw#/Port#  Port     Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status             Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------   --------  --------  ----  ------  ----  ---------  --------
    1/1       OK        2      50 cm     Yes    Yes    Yes       1          No
    1/2      Down      None    50 cm      No     No     No       2          No
    2/1      Down      None    50 cm      No     No     No       2          No
    2/2       OK        1      50 cm     Yes    Yes    Yes       1          No
```

Diagnosing the problem:

- The Stack Port Status value is *Down*.
- Link OK, Link Active, and Sync OK values are *No*.
- The Cable Length value is *50 cm*. The switch detects and correctly identifies the cable.

The connection between Port 2 on Switch 1 and Port 1 on Switch 2 is unreliable on at least one of the connector pins.

Additional References for Switch Stacks

Related Documents

Related Topic	Document Title
Cabling and powering on a switch stack.	*Catalyst 3850 Switch Hardware Installation Guide*
SGACL High Availability	"Cisco TrustSec SGACL High Availability" module of the *Cisco TrustSec Switch Configuration Guide*

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

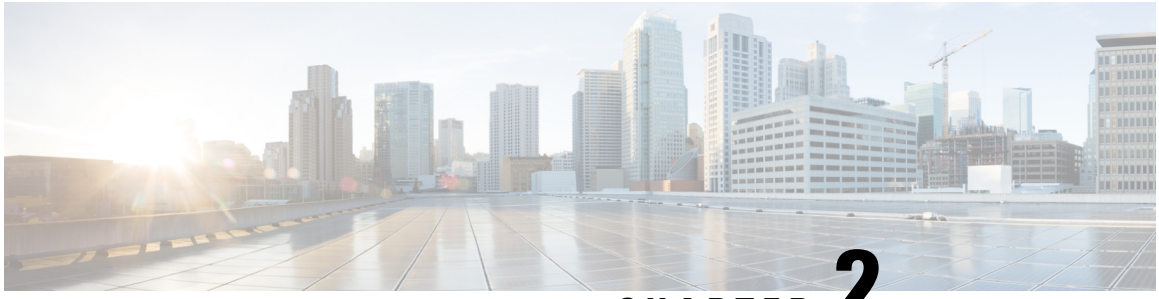
Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and licensed feature sets,, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support



CHAPTER 2

Configuring Cisco NSF with SSO

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NSF with SSO

The following are prerequisites and considerations for configuring NSF with SSO.

- Use of the routing protocols requires the IP Services license level. EIGRP-stub and OSPF for routed access are supported on IP Base license level.
- BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, the devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable router discovers that a particular BGP neighbor does not have graceful restart capability, it does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability continue to have NSF-capable sessions with this NSF-capable networking device.
- OSPF support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers that it has non-NSF -aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers continue to provide NSF capabilities.

Restrictions for NSF with SSO

The following are restrictions for configuring NSF with SSO:

- NSF does not support IP Multicast Routing, as it is not SSO-aware.
- NSF is not supported if the IOS-XE software is running in the LAN Base mode.
- For NSF operation, you must have SSO configured on the device.
- All Layer 3 neighboring devices must be NSF Helper or NSF-capable to support graceful restart capability.
- For IETF, all neighboring devices must be running an NSF-aware software image.

Information About NSF with SSO

Overview of NSF with SSO

The switch supports fault resistance by allowing a standby switch to take over if the active switch becomes unavailable. Cisco nonstop forwarding (NSF) works with stateful switchover (SSO) to minimize the amount of time a network is unavailable.

NSF provides these benefits:

- Improved network availability—NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.
- Overall network stability—Network stability may be improved with the reduction in the number of route flaps, which were created when routers in the network failed and lost their routing tables.
- Neighboring routers do not detect a link flap—Because the interfaces remain up during a switchover, neighboring routers do not detect a link flap (the link does not go down and come back up).
- Prevents routing flaps—Because SSO continues forwarding network traffic during a switchover, routing flaps are avoided.
- Maintains user sessions established prior to the switchover.

Keepalive messages are sent and received between the active and standby switches.

- If the standby switch does not respond, a new standby switch is elected.
- If the active switch does not respond, the standby switch becomes the active switch.

In addition, hello messages are sent and received by all stack members.

- If a stack member does not respond, that member is removed from the stack.
- If the standby switch does not respond, a new standby switch is elected.
- If the active switch does not respond, the standby switch becomes the active switch.

SSO Operation

When a standby switch runs in SSO mode, the standby switch starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration of the active switch. It subsequently maintains the state on the protocols listed below, and all changes in hardware and software states for features that support stateful switchover are kept in synchronization. Consequently, it offers minimum interruption to Layer 2 sessions in a redundant active switch configuration.

If the active switch fails, the standby switch becomes the active switch. This new active switch uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding will be delayed until the routing tables have been repopulated in the newly active switch.

Note SSO Layer 2 Only is supported if the IOS-XE software is running the LAN Base license level.

The state of these features is preserved between both the active and standby switches:

- 802.3
- 802.3u
- 802.3x (Flow Control)
- 802.3ab (GE)
- 802.3z (Gigabit Ethernet including CWDM)
- 802.3ad (LACP)
- 802.1p (Layer 2 QoS)
- 802.1q
- 802.1X (Authentication)
- 802.1D (Spanning Tree Protocol)
- 802.3af (Inline power)
- PAgP
- VTP
- Dynamic ARP Inspection
- DHCP
- DHCP snooping
- IP source guard
- IGMP snooping (versions 1 and 2)
- DTP (802.1q and ISL)
- MST
- PVST+

- Rapid-PVST
- PortFast/UplinkFast/BackboneFast
- BPDU guard and filtering
- Voice VLAN
- Port security
- Unicast MAC filtering
- ACL (VACLS, PACLS, RACLS)
- QOS (DBL)
- Multicast storm control/broadcast storm control

SSO is compatible with the following list of features. However, the protocol database for these features is not synchronized between the standby and active switches:

- 802.1Q tunneling with Layer 2 Protocol Tunneling (L2PT)
- Baby giants
- Jumbo frame support
- CDP
- Flood blocking
- UDLD
- SPAN/RSPAN
- NetFlow

All Layer 3 protocols on a switch are learned on the standby switch if SSO is enabled.

NSF Operation

Cisco IOS Nonstop Forwarding (NSF) always runs with stateful switchover (SSO) and provides redundancy for Layer 3 traffic. NSF is supported by the BGP, OSPF, and EIGRP routing protocols and is supported by Cisco Express Forwarding (CEF) for forwarding. The routing protocols have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. After the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the hardware with the new FIB information.

If the active switch is configured for BGP (with the **graceful-restart** command), OSPF, or EIGRP routing protocols, routing updates are automatically sent during the active switch election.

The switch supports NSF-awareness and NSF-capability for the BGP, OSPF, and EIGRP protocols in IP Services license level and NSF-awareness for the EIGRP-stub in IP Base license level.

NSF has two primary components:

- NSF-awareness

  A networking device is NSF-aware if it is running NSF-compatible software. If neighboring router devices detect that an NSF router can still forward packets when an active switch election happens, this capability is referred to as NSF-awareness. Cisco IOS enhancements to the Layer 3 routing protocols (BGP, OSPF, and EIGRP) are designed to prevent route-flapping so that the CEF routing table does not time out or the NSF router does not drop routes. An NSF-aware router helps to send routing protocol information to the neighboring NSF router. NSF-awareness is enabled by default for EIGRP-stub, EIGRP, and OSPF protocols. NSF-awareness is disabled by default for BGP.

- NSF-capability

  A device is NSF-capable if it has been configured to support NSF; it rebuilds routing information from NSF-aware or NSF-capable neighbors. NSF works with SSO to minimize the amount of time that a Layer 3 network is unavailable following an active switch election by continuing to forward IP packets. Reconvergence of Layer 3 routing protocols (BGP, OSPFv2, and EIGRP) is transparent to the user and happens automatically in the background. The routing protocols recover routing information from neighbor devices and rebuild the Cisco Express Forwarding (CEF) table.

Cisco Express Forwarding

A key element of Cisco IOS Nonstop Forwarding (NSF) is packet forwarding. In a Cisco networking device, packet forwarding is provided by Cisco Express Forwarding (CEF). CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active supervisor switch synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby switch. Upon switchover, the standby switch initially has FIB and adjacency databases that are mirror images of those that were current on the active switch. CEF keeps the forwarding engine on the standby switch current with changes that are sent to it by CEF on the active switch. The forwarding engine can continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates cause prefix-by-prefix updates to CEF, which it uses to update the FIB and adjacency databases. Existing and new entries receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the forwarding engine during convergence. The switch signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

BGP Operation

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a statement that the NSF-capable device has “graceful” restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peers need to exchange the graceful restart capability in their OPEN messages at the time of session establishment. If both the peers do not exchange the graceful restart capability, the session will not be capable of a graceful restart.

If the BGP session is lost during the active switch switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding

decisions for a set period of time. This functionality prevents packets from being lost while the newly active switch is waiting for convergence of the routing information with the BGP peers.

After an active switch switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. After this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table; the BGP protocol then is fully converged.

If a BGP peer does not support the graceful restart capability, it ignores the graceful restart capability in an OPEN message but establishes a BGP session with the NSF-capable device. This function allows interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers is not capable of a graceful restart.

Note BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, the devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable router discovers that a particular BGP neighbor does not have graceful restart capability, it does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability continue to have NSF-capable sessions with this NSF-capable networking device.

OSPF Operation

When an OSPF NSF-capable router performs an active switch switchover, it must perform the following tasks in order to resynchronize its link state database with its OSPF neighbors:

- Relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship
- Reacquire the contents of the link state database for the network

As quickly as possible after an active switch switchover, the NSF-capable router sends an OSPF NSF signal to neighboring NSF-aware devices. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this router should not be reset. As the NSF-capable router receives signals from other routers on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable router begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. Once this exchange is complete, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. The OSPF protocols are then fully converged.

Note OSPF support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers that it has non-NSF -aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers continue to provide NSF capabilities.

EIGRP Operation

When an EIGRP NSF-capable router initially re-boots after an NSF restart, it has no neighbor and its topology table is empty. The router is notified by the standby (now active) switch when it needs to bring up the interfaces, reacquire neighbors, and rebuild the topology and routing tables. The restarting router and its peers must accomplish these tasks without interrupting the data traffic directed toward the restarting router. EIGRP peer routers maintain the routes learned from the restarting router and continue forwarding traffic through the NSF restart process.

To prevent an adjacency reset by the neighbors, the restarting router uses a new Restart (RS) bit in the EIGRP packet header to indicate a restart. The RS bit is set in the hello packets and in the initial INIT update packets during the NSF restart period. The RS bit in the hello packets allows the neighbors to be quickly notified of the NSF restart. Without seeing the RS bit, the neighbor can only detect an adjacency reset by receiving an INIT update or by the expiration of the hello hold timer. Without the RS bit, a neighbor does not know if the adjacency reset should be handled using NSF or the normal startup method.

When the neighbor receives the restart indication, either by receiving the hello packet or the INIT packet, it recognizes the restarting peer in its peer list and maintains the adjacency with the restarting router. The neighbor then sends it topology table to the restarting router with the RS bit set in the first update packet indicating that it is NSF-aware and is helping out the restarting router. The neighbor does not set the RS bit in their hello packets, unless it is also a NSF restarting neighbor.

Note A router may be NSF-aware but may not be helping the NSF restarting neighbor because booting from a cold start.

If at least one of the peer routers is NSF-aware, the restarting router would then receive updates and rebuild its database. The restarting router must then find out if it had converged so that it can notify the routing information base (RIB). Each NSF-aware router is required to send an end of table (EOT) marker in the last update packet to indicate the end of the table content. The restarting router knows it has converged when it receives the EOT marker. The restarting router can then begin sending updates.

An NSF-aware peer would know when the restarting router had converged when it receives an EOT indication from the restarting router. The peer then scans its topology table to search for the routes with the restarted neighbor as the source. The peer compares the route timestamp with the restart event timestamp to determine if the route is still available. The peer then goes active to find alternate paths for the routes that are no longer available through the restarted router.

When the restarting router has received all EOT indications from its neighbors or when the NSF converge timer expires, EIGRP notifies the RIB of convergence. EIGRP waits for the RIB convergence signal and then floods its topology table to all awaiting NSF-aware peers.

How to Configure Cisco NSF with SSO

Configuring SSO

You must configure SSO in order to use NSF with any supported protocol.

SUMMARY STEPS

1. **redundancy**
2. **mode sso**
3. **end**
4. **show running-config**
5. **show redundancy states**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 2	**mode sso** **Example:** `Device(config-red)# mode sso`	Configures SSO. When this command is entered, the standby switch is reloaded and begins to work in SSO mode.
Step 3	**end** **Example:** `Device(config-red)# end`	Returns to EXEC mode.
Step 4	**show running-config** **Example:** `Device# show running-config`	Verifies that SSO is enabled.
Step 5	**show redundancy states** **Example:** `Device# show redundancy states`	Displays the operating redundancy mode.

Configuring SSO Example

This example shows how to configure the system for SSO and display the redundancy state:

```
Device(config)# redundancy
  Device(config)# mode sso
  Device(config)# end
  Device# show redundancy states
  my state = 13 -ACTIVE
  peer state = 8 -STANDBY HOT
  Mode = Duplex
  Unit = Primary
  Unit ID = 5
  Redundancy Mode (Operational) = sso
  Redundancy Mode (Configured) = sso
  Split Mode = Disabled
  Manual Swact = Enabled
  Communications = Up
  client count = 29
  client_notification_TMR = 30000 milliseconds
```

```
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0x0
```

Verifying CEF NSF

To verify CEF NSF, use the **show cef state** privileged EXEC command.

```
Device# show cef state
CEF Status:
RP instance
common CEF enabled
IPv4 CEF Status:
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
universal per-destination load sharing algorithm, id DEA83012
IPv6 CEF Status:
CEF disabled/not running
dCEF disabled/not running
universal per-destination load sharing algorithm, id DEA83012
RRP state:
I am standby RRP: no
RF Peer Presence: yes
RF PeerComm reached: yes
RF Progression blocked: never
Redundancy mode: rpr(1)
CEF NSF sync: disabled/not running
CEF ISSU Status:
FIBHWIDB broker
No slots are ISSU capable.
FIBIDB broker
No slots are ISSU capable.
FIBHWIDB Subblock broker
No slots are ISSU capable.
FIBIDB Subblock broker
No slots are ISSU capable.
Adjacency update
No slots are ISSU capable.
IPv4 table broker
No slots are ISSU capable.
CEF push
No slots are ISSU capable.
```

Configuring BGP for NSF

You must configure BGP graceful restart on all peer devices participating in BGP NSF.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp as-number**
3. **bgp graceful-restart**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 2	**router bgp as-number** **Example:** `Device(config)# router bgp 300`	Enables a BGP routing process, which places the switch in switch configuration mode.
Step 3	**bgp graceful-restart** **Example:** `Device(config)# bgp graceful-restart`	Enables the BGP graceful restart capability, starting BGP NSF. If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. Use this command on the restarting switch and all of its peers.

Verifying BGP NSF

To verify BGP NSF, you must check that BGP graceful restart is configured on the SSO-enabled networking device and on the neighbor devices. To verify, follow these steps:

Step 1 Verify that "bgp graceful-restart" appears in the BGP configuration of the SSO-enabled switch by entering the **show running-config** command:

Example:

```
Device# show running-config
.
.
.
router bgp 120
.
.
.
bgp graceful-restart
neighbor 192.0.2.0 remote-as 300
.
.
.
```

Step 2 Repeat Step 1 on each of the BGP neighbors.

Step 3 On the SSO device and the neighbor device, verify that the graceful restart function is shown as both advertised and received, and confirm the address families that have the graceful restart capability. If no address families are listed, BGP NSF does not occur either:

Example:

```
Device# show ip bgp neighbors
BGP neighbor is 192.0.2.3, remote AS 1, internal link
BGP version 4, remote router ID 192.0.2.4
BGP state = Established, up for 00:02:38
Last read 00:00:38, last write 00:00:35, hold time is 180, keepalive interval is 60
seconds
```

```
Neighbor capabilities:
Route refresh: advertised and received(new)
Address family IPv4 Unicast: advertised and received
Message statistics:
InQ depth is 0
OutQ depth is 0
Sent Rcvd
Opens: 1 1
Notifications: 0 0
Updates: 0 0
Keepalives: 4 4
Route Refresh: 0 0
Total: 5 5
Default minimum time between advertisement runs is 0 seconds
.............................................................
(Remaining output deleted)
```

Configuring OSPF NSF

All peer devices participating in OSPF NSF must be made OSPF NSF-aware, which happens automatically when you install an NSF software image on the device.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *processID*
3. **nsf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 2	**router ospf** *processID* **Example:** `Device(config)# router ospf processID`	Enables an OSPF routing process, which places the switch in router configuration mode.
Step 3	**nsf** **Example:** `Device(config)# nsf`	Enables NSF operations for OSPF.

Verifying OSPF NSF

Step 1 Verify that 'nsf' appears in the OSPF configuration of the SSO-enabled device by entering the show running-config command:

Example:

```
Device(config)#show running-config
route ospf 120
log-adjacency-changes
nsf
network 192.0.2.0 192.0.2.255 area 0
network 192.0.2.1 192.0.2.255 area 1
network 192.0.2.2 192.0.2.255 area 2
.
.
.
```

Step 2 Enter the **show ip ospf** command to verify that NSF is enabled on the device:

Example:

```
Device show ip ospf
Routing Process "ospf 1" with ID 192.0.2.1
Start time: 00:02:07.532, Time elapsed: 00:39:05.052
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
transit capable is 0
External flood list length 0
IETF Non-Stop Forwarding enabled
restart-interval limit: 120 sec
IETF NSF helper support enabled
Cisco NSF helper support enabled
Reference bandwidth unit is 100 mbps
Area BACKBONE(0)
Number of interfaces in this area is 3 (1 loopback)
Area has no authentication
SPF algorithm last executed 00:08:53.760 ago
SPF algorithm executed 2 times
Area ranges are
Number of LSA 3. Checksum Sum 0x025BE0
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

Configuring EIGRP NSF

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *as-number*
3. **nsf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device configure terminal`	Enters global configuration mode.
Step 2	**router eigrp** *as-number* **Example:** `Device(config)# router eigrp as-number`	Enables an EIGRP routing process, which places the switch in router configuration mode.
Step 3	**nsf** **Example:** `Device(config-router)# nsf`	Enables EIGRP NSF. Use this command on the "restarting" switch and all of its peers.

Verifying EIGRP NSF

Step 1 Verify that "nsf" appears in the EIGRP configuration of the SSO-enabled device by entering the show **running-config command**:

Example:

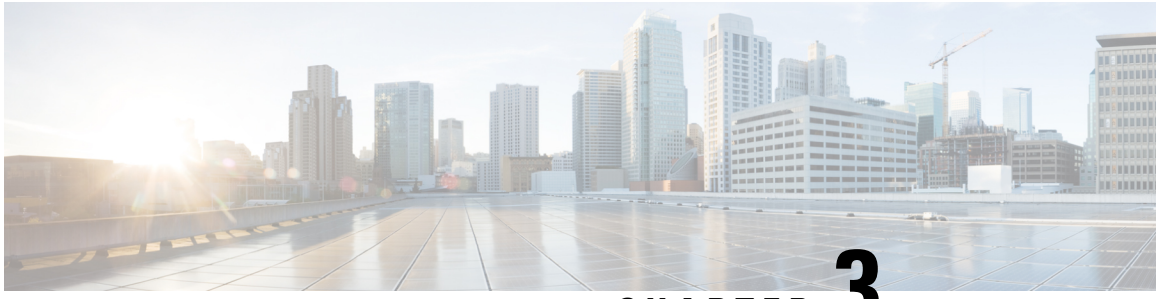
```
Device show running-config
..
.
router eigrp 100
auto-summary
nsf
..
.
```

Step 2 Enter the **show ip protocols** command to verify that NSF is enabled on the device:

Example:

```
Device show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.0.2.3
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Maximum path: 1
Routing for Networks:
Routing on Interfaces Configured Explicitly (Area 0):
Loopback0
GigabitEthernet5/3
TenGigabitEthernet3/1
Routing Information Sources:
Gateway Distance Last Update
192.0.2.1 110 00:01:02
Distance: (default is 110)
Routing Protocol is "bgp 601"
Outgoing update filter list for all interfaces is not set
```

```
Incoming update filter list for all interfaces is not set
IGP synchronization is disabled
Automatic route summarization is disabled
Neighbor(s):
Address FiltIn FiltOut DistIn DistOut Weight RouteMap
192.0.2.0
Maximum path: 1
Routing Information Sources:
Gateway Distance Last Update
192.0.2.0 20 00:01:03
Distance: external 20 internal 200 local 200
```



CHAPTER 3

Configuring Cisco StackWise Virtual

Prerequisites for Cisco StackWise Virtual

- All the switches in Cisco StackWise Virtual solution must be of the same switch model.
- All the switches in Cisco StackWise Virtual solution must be running the same license level.
- All the switches in Cisco StackWise Virtual must be running the same software version.

Note When you enable Cisco StackWise Virtual on the device:

- Layer 2, Layer 3, Security, Quality of Service, Multicast, Application, Monitoring and Management, Multiprotocol Label Switching, and High Availability are supported. Contact the Cisco Technical Support Centre for the specific list of features that are supported under each one of these technologies.
- Resilient Ethernet Protocol, Remote Switched Port Analyzer, and Sofware-Defined Access are NOT supported.

Restrictions for Cisco StackWise Virtual

- The Federal Information Processing Standards (FIPS) is not supported on Cisco StackWise Virtual links (SVLs).
- Cisco StackWise Virtual is supported only on the following Cisco Catalyst 3850 Series Switch models.
  - WS-C3850-24XS-S

- WS-C3850-24XS-E
- WS-C3850-12XS-S
- WS-C3850-12XS-E
- WS-C3850-48XS-S
- WS-C3850-48XS-E
- WS-C3850-48XS-F-S
- WS-C3850-48XS-F-E

- The dual-active and SVL configuration are performed manually and the device should be rebooted for the configuration changes to take effect.
- Cisco StackWise Virtual is supported in IP services and IP base licenses.The licenses must be matched between two Cisco StackWise Virtual member switches. License migration requires both Cisco StackWise Virtual member switches to be rebooted to activate the licenses.
- When deploying Cisco StackWise Virtual, ensure that VLAN ID 4094 is not used anywhere on the network. All inter-chassis system control communication between stack members is carried over the reserved VLAN ID 4094 from the global range.
- 4x10G break-out cables are not supported with SVLs.
- Configuring SVLs on any of the network modules is not supported.
- Dual-Active Detection is not supported on the uplink ports of the switch.

Information About Cisco Stackwise Virtual

StackWise Virtual Overview

Cisco StackWise Virtual is a network system virtualization technology that pairs two switches into one virtual switch. Switches in a Cisco StackWise Virtual solution simplify operational efficiency with a single control and management plane, scale system bandwidth with distributed forwarding plane, and assist in building resilient networks using the recommended network design. Cisco StackWise Virtual allows two physical switches to operate as a single logical virtual switch using a 40G or 10G Ethernet connection.

Cisco StackWise Virtual Topology

A typical network design consists of core, distribution, and access layers. The default mode of a switch is standalone. When two redundant switches are deployed in the distribution layer, the following network challenges arise:

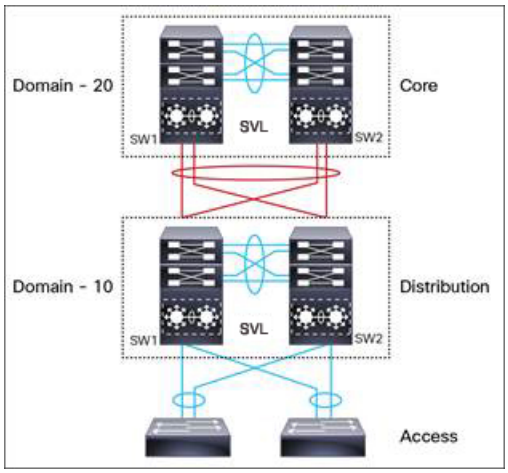
- If VLAN IDs are reused between access layers then, it will introduce a spanning tree loop that will impact the overall performance of the network.
- Spanning tree protcols and configuration are required to protect Layer 2 network against spanning tree protcol loop, and root and bridge protocol data unit management.

- Additional protocols such as first hop redundancy protocol are required to virtualize the IP gateway function. This should align with STP root priorities for each VLAN.
- The Protocol independent multicast designated router (PIM DR) configuration should be fine-tuned to selectively build a multicast forwarding topology on a VLAN.
- The standalone distribution layer system provides protocol-driven remote failure and detection, which results in slower convergence time. Fine-tune FHRP and PIM timers for rapid fault detection and recovery process.

We recommend the Cisco StackWise Virtual model for aggregation layers and collapsed aggregation and core layers. The stack can be formed over a redundant 40G or 10G fiber links to ensure that the distribution or the aggregation switches can be deployed over a large distance.

Note that STP keeps one of the ports connected to the distribution switches blocked on the access switches. As a result of this, an active link failure causes STP convergence and the network suffers from traffic loss, flooding, and a possible transient loop in the network. On the other hand, if the switches are logically merged into one switch, all the access switches might form an EtherChannel bundle with distribution switches, and a link failure within an EtherChannel would not have any impact as long as at least one member within the EtherChannel is active.

Figure 1: Typical Network Design using Cisco StackWise Virtual



Etherchannel in StackWise Virtual is capable of implementing MEC across the stack members. When access layer and aggregation layer are collapsed into a single StackWise Virtual system, MEC across the different access layer domain members and across distribution and access layer switches will not be supported. MEC is designed to forward the traffic over the local link irrespective of the hash result.

Since the control plane, management plane, and data plane are integrated, the system behaves as a single switch.

The virtualization of multiple physical switches into a single logical switch is from a control and management plane perspective only. Because of the control plane being common, it may look like a single logical entity to peer switches. The data plane of the switches are distributed. Each switch is capable of forwarding over its local interfaces without involving other members. However, when a packet coming into a switch has to be forwarded over a different member's port, the forwarding context of the packet is carried over to the destination switch after ingress processing is performed in the ingress switch. Egress processing is done only in the egress switch. This provides a uniform data plane behavior to the entire switch irrespective whether of the destination

port is in a local switch or in a remote switch. However, the common control plane ensures that all the switches have equivalent data plane entry for each forwarding entity.

An election mechanism elects one of the switches to be Cisco StackWise Virtual active and the other switch to be Cisco StackWise Virtual standby in terms of Control Plane functions. The active switch is responsible for all the management, bridging and routing protocols, and software data path. The standby switch is in hot standby state ready to take over the role of active, if the active switch fails over.

The following are the components of the Cisco StackWise Virtual solution:

- Stack members
- StackWise Virtual link: 10G or 40G Ethernet connections

StackWise Virtual link is the link that connects the switches over Ethernet. Typically, Cisco StackWise Virtual consits of multiple 10G or 40G physical links. It carries all the control and data traffic between the switching units. You can configure a StackWise Virtual link on any supported port. When a switch is powered up and the hardware is initiliazed, it looks for a configured StackWise Virtual link before the initilization of the control plane.

The Link Management Protocol (LMP) is activated on each link of the StackWise Virtual links as soon as the links are established. LMP ensure the integrity of SVL links and monitors and maintains the health of the links. The redundancy role of each switch is resolved by the StackWise Discovery Protocol (SDP). It ensures that the hardware and software versions are compatible to form the SVL and determines which switch becomes active or standby from a control plane perspective.

Cisco StackWise Virtual Header (SVH) is 64-byte overhead frame that is appended over all control, data, and management plane traffic that traverse over each SVL between the two stack members of the Cisco StackWise Virtual domain. The SVH-encapsulated traffic operates at OSI Layer 2 and can be recognized and processed only by Cisco StackWise Virtual-enabled switches. SVL interfaces are nonbridgeable, and allows nonrouteable traffic over a L2 or L3 network.

Cisco StackWise Virtual Redundancy

Cisco StackWise Virtual operates stateful switchover (SSO) between the active and standby switches. The following are the ways in which Cisco StackWise Virtual's redundancy model differs from that of the standalone mode:

- The Cisco StackWise Virtual active and standby switches are hosted in separate switches and use a StackWise Virtual link to exchange information.
- The active switch controls both the switches of Cisco StackWise Virtual. The active switch runs the Layer 2 and Layer 3 control protocols and manages the switching modules of both the switches.
- The Cisco StackWise Virtual active and standby switches perform data traffic forwarding.

Note If the Cisco StackWise Virtual active switch fails, the standby switch initiates a switchover and assumes the Cisco StackWise Virtual active switch role.

SSO Redundancy

A StackWise Virtual system operates with SSO redundancy if it meets the following requirements:

- Both the switches must be running the same software version, unless they are in the process of software upgrade.
- StackWise Virtual link-related configuration in the two switches must match.
- License type must be same on both the switch models.
- Both the switch models must be in the same StackWise Virtual domain.

With SSO redundancy, the StackWise Virtual standby switch is always ready to assume control if a fault occurs on the StackWise Virtual active switch. Configuration, forwarding, and state information are synchronized from the StackWise Virtual active switch to the redundant switch at startup, and whenever changes to the StackWise Virtual active switch configuration occur. If a switchover occurs, traffic disruption is minimized.

If StackWise Virtual does not meet the requirements for SSO redundancy, it will be incapable of establishing a relationship with the peer switch. StackWise Virtual runs stateful switchover (SSO) between the StackWise Virtual active and standby switches. The StackWise Virtual determines the role of each switch during initialization.

The CPU in the StackWise Virtual standby switch runs in hot standby state. StackWise Virtual uses a StackWise Virtual link to synchronize configuration data from the StackWise Virtual active switch to the StackWise Virtual standby switch. Also, protocols and features that support high availability synchronize their events and state information to the StackWise Virtual standby switch.

Nonstop Forwarding

While implementing Nonstop Forwarding (NSF) technology in systems using SSO redundancy mode, network disruptions are transparent to campus users and applications. High availability is provided even when the control-plane processing stack-member switch is reset. During a failure of the underlying Layer 3, NSF-capable protocols perform graceful network topology resynchronization. The preset forwarding information on the redundant stack-member switch remains intact; this switch continues to forward the data in the network. This service availability significantly lowers the mean time to repair (MTTR) and increases the mean time between failure (MTBF) to achieve a high level of network availability.

Multichassis EtherChannels

A Multichassis EtherChannel (MEC) is an EtherChannel bundled with physical ports having common characteristics such as speed and duplex, that are distributed across each Cisco StackWise Virtual system. A Cisco StackWise Virtual MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch). Cisco StackWise Virtual supports up to 128 MECs deployed in Layer 2 or Layer 3 modes. EtherChannel 128 is reserved for SVL connections. Hence, the maximum available MEC count is 127.

In a Cisco StackWise Virtual system, an MEC is an EtherChannel with additional capability. A multichassis EtherChannel link reduces the amount of traffic that requires transmission across the StackWise Virtual link by populating the index port only with the ports local to the physical switch. This allows the switch to give precedence to the local ports of the multichassis EtherChannel link over those on the remote switch.

Each MEC can optionally be configured to support either Cisco PAgP, IEEE LACP, or Static ON mode. We recommend that you implement EtherChannel using Cisco PAgP or LACP with a compatible neighbor. If a remotely connected neighbor such as Cisco Wireless LAN Controller (WLC) does not support this link-bundling protocol, then a Static ON mode can be deployed. These protocols run only on the Cisco StackWise Virtual active switch.

An MEC can support up to eight physical links that can be distributed in any proportion between the Cisco StackWise Virtual active switch and the Cisco StackWise Virtual standby switch. We recommend that you distribute the MEC ports across both switches evenly.

MEC Minimum Latency Load Balancing

The StackWise Virtual environment is designed such that data forwarding always remains within the switch. The Virtual Stack always tries to forward traffic on the locally available links. This is true for both Layer 2 and Layer3 links. The primary motivation for local forwarding is to avoid unnecessarily sending data traffic over the StackWise Virtual link and thus reduce the latency (extra hop over the SVL) and congestion.The bidirectional traffic is load-shared between the two StackWise Virtual members. However, for each StackWise Virtual member, ingress and egress traffic forwarding is based on locally-attached links that are part of MEC. This local forwarding is a key concept in understanding convergence and fault conditions in a StackWise Virtual enabled campus network.

The active and standby switches support local forwarding that will individually perform the desired lookups and forward the traffic on local links to uplink neighbors. If the destination is a remote switch in the StackWise Virtual domain, ingress processing is performed on the ingress switch and then traffic is forwarded over the StackWise Virtual link to the egress switch where only egress processing is performed.

MEC Failure Scenarios

We recommend that you configure a MEC with at least one link to each switch. This configuration ensures that there is always an alternate path for data traffic in case of a switch failure.

The following sections describe issues that may arise and the resulting impact:

Single MEC Link Failure

If a link within a MEC fails (and other links in the MEC are still operational), the MEC redistributes the load among the operational links, as in a regular port.

All MEC Links to the Cisco StackWise Virtual Active Switch Fail

If all the links to the Cisco StackWise Virtual active switch fail, a MEC becomes a regular EtherChannel with operational links to the Cisco StackWise Virtual standby switch.

Data traffic that terminates on the Cisco StackWise Virtual active switch reaches the MEC by crossing a StackWise Virtual link to the Cisco StackWise Virtual standby switch. Control protocols continue to run in the Cisco StackWise Virtual active switch. Protocol messages reach the MEC by crossing a StackWise Virtual link.

All MEC Links Fail

If all the links in an MEC fail, the logical interface for the EtherChannel is set to Unavailable. Layer 2 control protocols perform the same corrective action as for a link-down event on a regular EtherChannel.

On adjacent switches, routing protocols and the Spanning Tree Protocol (STP) perform the same corrective action as for a regular EtherChannel.

Cisco StackWise Virtual Standby Switch Failure

If the Cisco StackWise Virtual standby switch fails, a MEC becomes a regular EtherChannel with operational links on the Cisco StackWise Virtual active switch. Connected peer switches detect the link failures, and adjust their load-balancing algorithms to use only the links to the StackwWise Virtual active switch.

Cisco StackWise Virtual Active Switch Failure

Cisco StackWise Virtual active switch failure results in a stateful switchover (SSO). After the switchover, a MEC is operational on the new Cisco StackWise Virtual active switch. Connected peer switches detect the link failures (to the failed switch), and adjust their load-balancing algorithms to use only the links to the new Cisco StackWise Virtual active switch.

Cisco StackWise Virtual Packet Handling

In Cisco StackWise Virtual, the Cisco StackWise Virtual active switch runs the Layer 2 and Layer 3 protocols and features and manages the ports on both the switches.

Cisco StackWise Virtual uses StackWise Virtual link to communicate system and protocol information between the peer switches and to carry data traffic between the two switches.

The following sections describe packet handling in Cisco StackWise Virtual.

Traffic on a StackWise Virtual link

A StackWise Virtual link carries data traffic and in-band control traffic between two switches. All the frames that are forwarded over the StackWise Virtual link are encapsulated with a special StackWise Virtual Header (SVH). The SVH adds an overhead of 64 bytes for control and data traffic, which provides information for Cisco StackWise Virtual to forward the packet on the peer switch.

A StackWise Virtual link transports control messages between two switches. Messages include protocol messages that are processed by the Cisco StackWise Virtual active switch, but received or transmitted by interfaces on the Cisco StackWise Virtual standby switch. Control traffic also includes module programming between the Cisco StackWise Virtual active switch and the switching modules on the Cisco StackWise Virtual standby switch.

Cisco StackWise Virtual transmits data traffic over a StackWise Virtual link under the following circumstances:

- Layer 2 traffic flooded over a VLAN (even for dual-homed links).
- Packets processed by software on the Cisco StackWise Virtual active switch where the ingress interface is on the Cisco StackWise Virtual standby switch.
- The packet destination is on the peer switch, as described in the following examples:
  - Traffic within a VLAN where the known destination interface is on the peer switch.
  - Traffic that is replicated for a multicast group and the multicast receivers are on the peer switch.
  - The known unicast destination MAC address is on the peer switch.
  - The packet is a MAC notification frame destined for a port on the peer switch.

A StackWise Virtual link also transports system data, such as NetFlow export data and SNMP data, from the Cisco StackWise Virtual standby switch to the Cisco StackWise Virtual active switch.

Traffic on the StackWise Virtual link is load balanced with the same global hashing algorithms available for EtherChannels (the default algorithm is source-destination IP).

Layer 2 Protocols

The Cisco StackWise Virtual active switch runs the Layer 2 protocols (such as STP and VTP) for the switching modules on both the switches. Protocol messages that are transmitted and received on the Cisco StackWise Virtual standby switch switching modules must traverse a StackWise Virtual link to reach the Cisco StackWise Virtual active switch.

All the Layer 2 protocols in Cisco StackWise Virtual work similarly in standalone mode. The following sections describe the difference in behavior for some protocols in Cisco StackWise Virtual.

Spanning Tree Protocol

The Cisco StackWise Virtual active switch runs the STP. The Cisco StackWise Virtual standby switch redirects the STP BPDUs across a StackWise Virtual link to the Stackwise Virtual active switch.

The STP bridge ID is commonly derived from the switch MAC address. To ensure that the bridge ID does not change after a switchover, Cisco StackWise Virtual continues to use the original switch MAC address for the STP Bridge ID.

EtherChannel Control Protocols

Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets contain a device identifier. Cisco StackWise Virtual defines a common device identifier for both the switches. Use either PAgP or LACP on Multi EtherChannels instead of mode ON, even if all the three modes are supported.

Note A new PAgP enhancement has been defined for assisting with dual-active scenario detection.

Switched Port Analyzer

Switched Port Analyzer (SPAN) on StackWise Virtual link ports is not supported; SVL ports can be neither a SPAN source, nor a SPAN destination. Cisco StackWise Virtual supports all the SPAN features for non-SVL interfaces. The number of SPAN sessions that are available on Cisco StackWise Virtual matches that on a single switch running in standalone mode.

Private VLANs

Private VLANs on Stackwise Virtual work the same way as in standalone mode. The only exception is that the native VLAN on isolated trunk ports must be configured explicitly.

Apart from STP, EtherChannel Control Protocols, SPAN, and private VLANs, the Dynamic Trunking Protocol (DTP), Cisco Discovery Protocol (CDP), VLAN Trunk Protocol (VTP), and Unidirectional Link Detection Protocol (UDLD) are the additional Layer 2 control-plane protocols that run over the SVL connections.

Layer 3 Protocols

The Cisco StackWise Virtual active switch runs the Layer 3 protocols and features for the Stackwise Virtual. All the Layer 3 protocol packets are sent to and processed by the Cisco StackWise Virtual active switch. Both the member switches perform hardware forwarding for ingress traffic on their interfaces. When software forwarding is required, packets are sent to the Cisco StackWise Virtual active switch for processing.

The same router MAC address assigned by the Cisco StackWise Virtual active switch is used for all the Layer 3 interfaces on both the Cisco StackWise Virtual member switches. After a switchover, the original router

MAC address is still used. The router MAC address is chosen based on chassis-mac and is preserved after switchover by default.

The following sections describe the Layer 3 protocols for Cisco StackWise Virtual.

IPv4 Unicast

The CPU on the Cisco StackWise Virtual active switch runs the IPv4 routing protocols and performs any required software forwarding. All the routing protocol packets received on the Cisco StackWise Virtual standby switch are redirected to the Cisco StackWise Virtual active switch across the StackWise Virtual link. The Cisco StackWise Virtual active switch generates all the routing protocol packets to be sent out over ports on either of the Cisco StackWise Virtual member switches.

Hardware forwarding is distributed across both members on Cisco StackWise Virtual. The CPU on the Cisco StackWise Virtual active switch sends Forwarding Information Base (FIB) updates to the Cisco StackWise Virtual standby switch, which in turn installs all the routes and adjacencies into hardware.

Packets intended for a local adjacency (reachable by local ports) are forwarded locally on the ingress switch. Packets intended for a remote adjacency (reachable by remote ports) must traverse the StackWise Virtual link.

The CPU on the Cisco StackWise Virtual active switch performs all software forwarding and feature processing (such as fragmentation and Time to Live exceed functions). If a switchover occurs, software forwarding is disrupted until the new Cisco StackWise Virtual active switch obtains the latest Cisco Express Forwarding and other forwarding information.

In virtual switch mode, the requirements to support non-stop forwarding (NSF) match those in the standalone redundant mode of operation.

From a routing peer perspective, Multi-Chassis EtherChannels (MEC) remain operational during a switchover, that is, only the links to the failed switch are down, but the routing adjacencies remain valid.

Cisco StackWise Virtual achieves Layer 3 load balancing over all the paths in the Forwarding Information Base entries, be it local or remote.

IPv6

Cisco StackWise Virtual supports IPv6 unicast and multicast because it is present in the standalone system.

IPv4 Multicast

The IPv4 multicast protocols run on the Cisco StackWise Virtual active switch. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the Cisco StackWise Virtual standby switch are transmitted across a StackWise Virtual link to the Stackwise Virtual active switch. The latter generates IGMP and PIM protocol packets to be sent over ports on either of the Cisco StackWise Virtual members.

The Cisco StackWise Virtual active switch synchronizes the Multicast Forwarding Information Base (MFIB) state to the Cisco StackWise Virtual standby switch. On both the member switches, all the multicast routes are loaded in the hardware, with replica expansion table (RET) entries programmed for only local, outgoing interfaces. Both the member switches are capable of performing hardware forwarding.

Note To avoid multicast route changes as a result of a switchover, we recommend that all the links carrying multicast traffic be configured as MEC rather than Equal Cost Multipath (ECMP).

For packets traversing a StackWise Virtual link, all Layer 3 multicast replications occur on the egress switch. If there are multiple receivers on the egress switch, only one packet is replicated and forwarded over the StackWise Virtual link, and then replicated to all the local egress ports.

Software Features

Software features run only on the Cisco StackWise Virtual active switch. Incoming packets to the Cisco StackWise Virtual standby switch that require software processing are sent across a StackWise Virtual link to the Cisco StackWise Virtual active switch.

Dual-Active Detection

If the standby switch detects a complete loss of the StackWise Virtual link, it assumes the active switch has failed and will take over as the active switch. However, if the original Cisco StackWise Virtual active switch is still operational, both the switches will now be Cisco StackWise Virtual active switches. This situation is called a dual-active scenario. This scenario can have adverse effects on network stability because both the switches use the same IP addresses, SSH keys, and STP bridge IDs. Cisco StackWise Virtual detects a dual-active scenario and takes recovery action. Dual-active-detection link is the dedicated link used to mitigate this.

If a StackWise Virtual link fails, the Cisco StackWise Virtual standby switch cannot determine the state of the Cisco StackWise Virtual active switch. To ensure that switchover occurs without delay, the Cisco StackWise Virtual standby switch assumes that the Cisco StackWise Virtual active switch has failed and initiates switchover to take over the Cisco StackWise Virtual active role. The original Cisco StackWise Virtual active switch enters recovery mode and brings down all the interfaces except the StackWise Virtual link and the management interfaces.

Dual-Active-Detection Link with Fast Hello

To use the dual-active fast hello packet detection method, you must provision a direct ethernet connection between the two Cisco StackWise Virtual switches. You can dedicate up to four links for this purpose.

The two switches periodically exchange special dual-active hello messages containing information about the switch state. If all Stackwise Virtual Links fail and a dual-active scenario occurs, each switch recognizes that there is a dual-active scenario from the peer's messages. This initiates recovery actions as described in the Recovery Actions, on page 57 section. If a switch does not receive an expected dual-active fast hello message from the peer before the timer expires, the switch assumes that the link is no longer capable of dual-active detection.

Note Do not use the same port for StackWise Virtual link and dual-active detection link.

Dual-Active Detection Using Enhanced PAgP

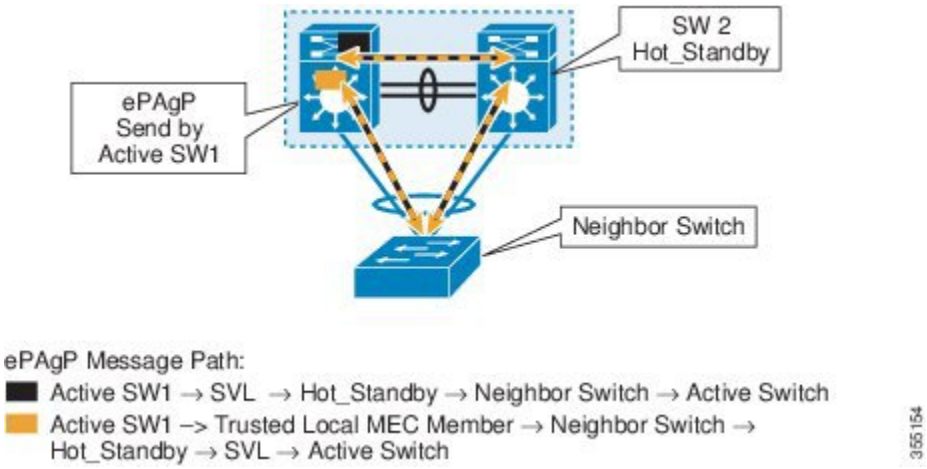
Port aggregation protocol (PAgP) is a Cisco-proprietary protocol for managing EtherChannels. If a Stackwise Virtual MEC terminates to a Cisco switch, you can run PAgP protocol on the MEC. If PAgP is running on the MECs between the Stackwise Virtual and an upstream or downstream switch, the Stackwise Virtual can use PAgP to detect a dual-active scenario. The MEC must have at least one port on each switch of the Stackwise Virtual.

Enhanced PAgP is an extension of the PAgP protocol. In virtual switch mode, PAgP messages include a new type length value (TLV) which contains the ID of the StackWise Virtual active switch. Only switches in virtual switch mode send the new TLV.

When the StackWise Virtual standby switch detects SVL failure, it initiates SSO and becomes StackWise Virtual active. Subsequent PAgP messages to the connected switch from the newly StackWise Virtual active switch contain the new StackWise Virtual active ID. The connected switch sends PAgP messages with the new StackWise Virtual active ID to both Stackwise Virtual switches.

If the formerly Stackwise Virtual active switch is still operational, it detects the dual-active scenario because the StackWise Virtual active ID in the PAgP messages changes.

Figure 2: Dual-active-detection with ePAgP



Note To avoid PAgP flaps and to ensure that dual-active detection functions as expected, the stack MAC persistency wait timer must be configured as indefinite using the command **stack-mac persistent timer 0 .**

Recovery Actions

A Cisco Stackwise Virtual active switch that detects a dual-active condition shuts down all of its non-StackWise Virtual Link interfaces to remove itself from the network. The switch then waits in recovery mode until the StackWise Virtual links have been recovered. You should physically repair the StackWise Virtual link failure and the recovery switch should be manually reloaded for it to be the standby switch.

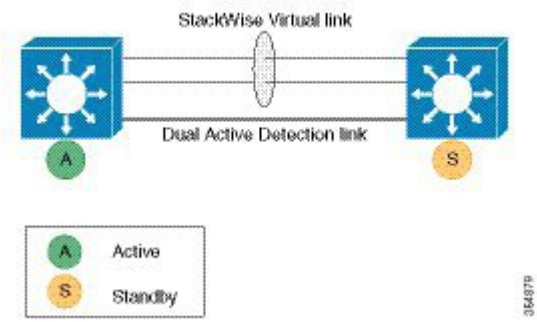
Implementing Cisco StackWise Virtual

The two-node solution of Cisco StackWise Virtual is normally deployed at the aggregation layer. Two switches are connected over a StackWise Virtual link (SVL).

Cisco StackWise Virtual combines the two switches into a single logical switch with a large number of ports, offering a single point of management. One of the member switches is the active and works as the control and management plane, while the other one is the standby. The virtualization of multiple physical switches into a single logical switch is only from a control and management perspective. Because of the control plane being common, it may look like a single logical entity to peer switches. The data plane of the switches are converged, that is, the forwarding context of a switch might be passed to the other member switch for further

processing when traffic is forwarded across the switches. However, the common control plane ensures that all the switches have equivalent data plane entry for each forwarding entity.

Figure 3: Two-Node Solution



An election mechanism that determines which switch has Cisco StackWise Virtual active and which one is a control plane standby, is available. The active switch is responsible for management, bridging and routing protocols, and software data path. These are centralized on the active switch supervisor of the Cisco StackWise Virtual active switch.

How to Configure Cisco StackWise Virtual

Configuring Cisco StackWise Virtual Settings

To enable StackWise Virtual, perform the following procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **stackwise-virtual**
4. **domain** *id*
5. **end**
6. **show stackwise-virtual**
7. **write memory**
8. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**stackwise-virtual** **Example:** `Device(config)# stackwise-virtual`	Enables Cisco StackWise Virtual and enters stackwise-virtual submode.
Step 4	**domain** *id* **Example:** `Device(config-stackwise-virtual)# domain 2`	(Optional) Specifies the Cisco StackWise Virtual domain ID. The domain ID range is from 1 to 255. The default value is one.
Step 5	**end** **Example:** `Device(config-stackwise-virtual)#end`	Returns to privileged EXEC mode.
Step 6	**show stackwise-virtual** **Example:** `Device#show stackwise-virtual`	
Step 7	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configurations for **stackwise-virtual** and **domain** are saved to the running-configuration and the startup-configuration after the reload.
Step 8	**reload** **Example:** `Device#reload`	Restarts the switch and forms the stack.

Configuring Cisco StackWise Virtual Link

Note Cisco StackWise Virtual link is supported on all 10G interfaces, and 40G interfaces. However, a combination of both interfaces is not supported.

To configure a 10 Gigabit Ethernet port as a StackWise Virtual link port, perform the following procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface TenGigabitEthernet** *<interface>*
4. **stackwise-virtual link** *link value*
5. **end**
6. **write memory**
7. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** *<interface>* **Example:** `Device(config)#interface TenGigabitEthernet1/0/2`	Enters 10G ethernet interface configuration mode. You can use the **interface FortyGigabitEthernet** *<interface>* command for a 40G ethernet interface.
Step 4	**stackwise-virtual link** *link value* **Example:** `Device(config-if)#stackwise-virtual link 1`	Associates the interface with configured StackWise Virtual link.
Step 5	**end** **Example:** `Device(config-if)#end`	Returns to privileged EXEC mode.
Step 6	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM to the startup-configuration in the system NVRAM. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **stackwise-virtual link** *link value* is saved only in the running-configuration and not the startup-configuration.
Step 7	**reload** **Example:** `Device#reload`	Restarts the switch.

Configuring a StackWise Virtual Dual-Active-Detection link

To configure a 10 Gigabit Ethernet port as StackWise dual-active-detection link, perform the following procedure. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface TenGigabitEthernet** *<interface>*
4. **stackwise-virtual dual-active-detection**
5. **end**
6. **write memory**
7. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** *<interface>* **Example:** `Device(config)#interface TenGigabitEthernet1/0/41`	Enters a 10G interface configuration mode. **Note** 1G interfaces can also be configured as Dual-active detection link. You must use different ports for StackWise Virtual link and dual-active detection link.
Step 4	**stackwise-virtual dual-active-detection** **Example:** `Device(config-if)#stackwise-virtual dual-active-detection`	Associates the interface with StackWise Virtual dual-active-detection. **Note** This command will not be visible on the device after the configuration, but will continue to function.
Step 5	**end** **Example:** `Device(config-if)#end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM to the startup-configuration in the system NVRAM. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **stackwise-virtual dual-active-detection** is saved only in the running-configuration and not the startup-configuration.
Step 7	**reload** **Example:** `Device#reload`	Restarts the switch.

Enabling ePAgP Dual-Active-Detection

To enable ePAgP dual-active-detection on a 10 Gigabit Ethernet, perform the following procedure. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface TenGigabitEthernet** *interface*
4. **channel-group** *group_ID* **mode desirable**
5. **exit**
6. **interface port-channel** *channel-group-id*
7. **shutdown**
8. **exit**
9. **stackwise-virtual**
10. **dual-active detection pagp**
11. **dual-active detection pagp trust channel-group** *channel-group id*
12. **exit**
13. **interface port-channel** *portchannel*
14. **no shutdown**
15. **end**
16. **write memory**
17. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** *interface* **Example:** `Device(config)#interface TenGigabitEthernet1/0/5`	Enters a 10G interface configuration mode.
Step 4	**channel-group** *group_ID* **mode desirable** **Example:** `Device(config-if)#channel-group 1 mode desirable`	Enables PAgP MEC with channel-group id in the range of 1 to 128 for 10GigabitEthernet interfaces.
Step 5	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 6	**interface port-channel** *channel-group-id* **Example:** `Device(config)#interface port-channel 1`	Selects a port channel interface to configure.
Step 7	**shutdown** **Example:** `Device(config-if)#shutdown`	Shuts down an interface.
Step 8	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 9	**stackwise-virtual** **Example:** `Device(config)#stackwise-virtual`	Enters the StackWise Virtual configuration mode.
Step 10	**dual-active detection pagp** **Example:** `Device(config-stackwise-virtual)#dual-active detection pagp`	Enables pagp dual-active detection. This is enabled by default.
Step 11	**dual-active detection pagp trust channel-group** *channel-group id* **Example:**	Enables dual-active detection trust mode on channel-group with the configured ID.

	Command or Action	Purpose
	`Device(config-stackwise-virtual)#dual-active detection pagp trust channel-group 1`	
Step 12	**exit** **Example:** `Device(config-stackwise-virtual)#exit`	Exits the StackWise-Virtual configuration mode.
Step 13	**interface port-channel** *portchannel* **Example:** `Device(config)#interface port-channel 1`	Configured port-channel on the switch.
Step 14	**no shutdown** **Example:** `Device(config-if)#no shutdown`	Enables the configured port-channel on the switch.
Step 15	**end** **Example:** `Device(config-if)#end`	Exits interface configuration.
Step 16	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **dual-active detection pagp trust channel-group** *channel-group id* is saved to the running-configuration and the startup-configuration after the reload.
Step 17	**reload** **Example:** `Device#reload`	Restarts the switch and configuration takes effect.

Disabling Cisco StackWise Virtual

To disable Cisco StackWise Virtual on a switch, perform the following procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface TenGigabitEthernet** *interface*
4. **no stackwise-virtual dual-active-detection**
5. **exit**
6. **interface TenGigabitEthernet** *interface*
7. **no stackwise-virtual link** *link*

8. **exit**
9. **no stackwise-virtual**
10. **exit**
11. **write memory**
12. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** *interface* **Example:** `Device(config)#interface TenGigabitEthernet1/0/41`	Enters a 10G interface configuration mode.
Step 4	**no stackwise-virtual dual-active-detection** **Example:** `Device(config-if)#no stackwise-virtual dual-active-detection`	Dissociates the 10G interface from StackWise Virtual dual-active-detection.
Step 5	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 6	**interface TenGigabitEthernet** *interface* **Example:** `Device(config)#interface TenGigabitEthernet1/0/5`	Enters a 10G interface configuration mode.
Step 7	**no stackwise-virtual link** *link* **Example:** `Device(config-if)#no stackwise-virtual link 1`	Dissociates the 10G interface from StackWise Virtual link.
Step 8	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.

	Command or Action	Purpose
Step 9	**no stackwise-virtual** **Example:** `Device(config)#no stackwise-virtual`	Disables StackWise Virtual configuration.
Step 10	**exit** **Example:** `Device(config)#exit`	Exits the global configuration mode.
Step 11	**write memory** **Example:** `Device#write memory`	Saves the running configuration. **Note** Starting Cisco IOS XE Everest 16.6.1, this step is optional.
Step 12	**reload** **Example:** `Device#reload`	Restarts the switch.

Verifying Cisco StackWise Virtual Configuration

To verify your Stackwise Virtual configuration, use the following **show** commands:

show stackwise-virtual switch *number <1-2>*	Displays information of a particular switch in the stack.
show stackwise-virtual link	Displays StackWise Virtual link information.
show stackwise-virtual bandwidth	Displays the bandwidth available for the Cisco StackWise Virtual.
show stackwise-virtual neighbors	Displays the Cisco StackWise Virtual neighbors.
show stackwise-virtual dual-active-detection	Displays Stackwise Virtual dual-active-detection information.
show stackwise-virtual dual-active-detection pagp	Displays ePAgP dual-active-detection information.

Additional References for StackWise Virtual

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Stack Manager and High Availability Command Reference

MIBs

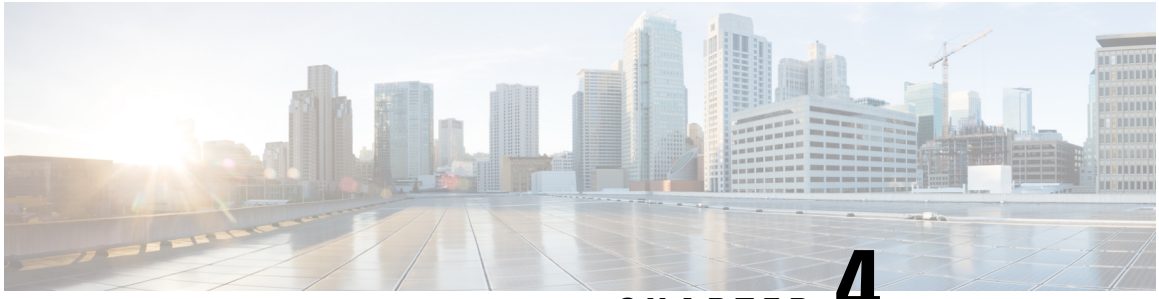
MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Cisco StackWise Virtual

Release	Modification
Cisco IOS XE Denali 16.3.3	This feature was introduced.
Cisco IOS XE Everest 16.5.1a	This feature was not supported.
Cisco IOS XE Everest 16.6.1	This feature was reintroduced. Minimum Latency Load Balancing and Dual-active-detection using ePAgP were introduced.



CHAPTER 4

Configuring 1:1 Redundancy

Prerequisites for 1:1 Redundancy

All the switches in the stack must be running the same license level as the active switch. For information about license levels, see the *System Management Configuration Guide.*

All the switches in the stack must be running compatible software versions.

Information About 1:1 Redundancy

1:1 redundancy is used to assign active and standby roles to specific switches in the stack. This overrides the traditional N+1 role selection algorithm, where any switch in the stack can be active or standby. In 1:1 redundancy, the stack manager determines the active and standby role for a specific switch, based on the flash ROMMON variable. The algorithm assigns one switch as active, another switch as standby, designating all remaining switches in the stack as members. When an active switch reboots it becomes standby and the existing standby switch will become active. The existing member switches remain in the same state.

How to Configure 1:1 Redundancy

Enabling 1:1 Redundancy Stack Mode

Follow these steps to enable the 1:1 redundancy stack mode, and set a switch as the active switch in a stack, or as the standby:

SUMMARY STEPS

1. **enable**
2. **switch** *switch-number* **role** {**active** | **standby**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**switch** *switch-number* **role** {**active**	**standby**} **Example:** `Device#switch 1 role active`	Changes stack mode to 1:1 mode and designates the switch as active or standby.

Disabling 1:1 Redundancy Stack Mode

On a switch where 1:1 redundancy is enabled, follow these steps to disable the feature. This changes the stack mode to N+1:

SUMMARY STEPS

1. **enable**
2. switch clear stack-mode

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	switch clear stack-mode **Example:** `Device# switch clear stack-mode`	Changes stack mode to the N+1 mode and removes active and standby assignments.

Verifying the Stack Mode

To verify the current stack mode on a switch, enter the **show switch stack-mode** command in priviledged EXEC mode. The output displays detailed status of the currently running stack mode.

```
Device# show switch stack-mode
Switch   Role    Mac Address    Version   Mode    Configured   State
-----------------------------------------------------------------
1       Member 3c5e.c357.c880             1+1'     Active'     Ready
*2      Active 547c.69de.cd00    V05      1+1'     Standby'    Ready
3       Member 547c.6965.cf80    V05      1+1'     Member'     Ready
```

The Mode field indicates the current stack mode

The Configured field refers to the switch state expected after a reboot.

Single quotation marks (') indicate that the stack mode has been changed.

Configuration Examples for 1:1 Redundancy

Example: Enabling 1:1 Redundancy Stack Mode

This example shows how to enable 1:1 redundancy stack mode. The stack will run in the 1:1 stack mode with the designated switches as active and standby, after reboot.

```
Device#
Device# Device 1 role active
WARNING: Changing the Device priority may result in a configuration change for that Device.
 Do you want to continue?[y/n]? [yes]: yes
Device#
Device# Device 2 role standby
WARNING: Changing the Device priority may result in a configuration change for that Device.
 Do you want to continue?[y/n]? [yes]: yes
Device#
```

Example: Disabling 1:1 Redundancy Stack Mode

This example shows how to disable 1:1 redundancy stack mode.

```
Device# switch clear stack-mode
WARNING: Changing the switch priority may result in a configuration change for that switch.
 Do you want to continue?[y/n]? [yes]: yes
Switch#
```

Additional References for 1:1 Redundancy

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Stack Manager and High Availability* section of the Command Reference guide for the release

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

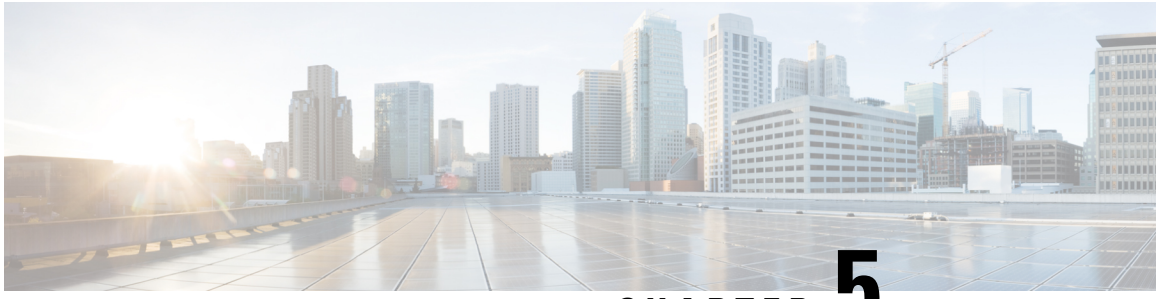
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for 1:1 Redundancy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Release	Modification
Cisco IOS XE Everest 16.6.1	This feature was introduced.



CHAPTER 5

Configuring ISSU

Prerequisites for Performing ISSU

The following prerequisites apply for performing ISSU:

- The active switch must have access to the new IOS XE image or pre-load it into flash.
- The switch must be running in install mode.
- Non-Stop Forwarding (NSF) must be enabled.

Information About ISSU Process

In-Service Software Upgrade (ISSU) is a process that upgrades an image to another image on a device while the network continues to forward packets. ISSU helps network administrators avoid a network outage when performing a software upgrade. The images are upgraded in install mode wherein each package is upgraded individually.

StackWise Virtual (SVL) comprises two switches that are connected together to form one virtual swtich. SVL supports In-Service Software Upgrades. ISSU on SVL is performed either in a single step or in three-steps.

ISSU supports upgrades, downgrades, and rollbacks.

For more information on configuring Stackwise Virtual, refer *Configuring Cisco Stackwise Virtual*.

ISSU Upgrade

The following steps describe the process that is followed in performing ISSU:

1. Copy the new image to the standby and active switches.
2. Unzip the files and copy packages to both the active and standby switches.

3. Install the packages on the standby switch.
4. Restart the standby switch.

   The standby switch is now upgraded to the new software.
5. Install the packages on the active switch.
6. Restart the active switch and switchover the standby to new active switch. After the switchover, the new standby switch will be up with the new software. The new software image is already installed on the new active switch, hence ISSU is completed.

ISSU Upgrade: 3-Step Work Flow

This workflow involves three steps—add, activate, and commit. After activation, all switches are upgraded to new software version except that the software is not committed automatically but must be performed manually via the **install commit** command. The advantage of this approach is the system can be rolled back to a previous software version. The system automatically rolls back if the rollback timer is not stopped using the **install abort-timer-stop** or the **install commit** command. If the rollback timer is stopped, the new software version could be run on the device for any duration and then rolled back to the previous version.

ISSU Upgrade: 1-Step Work Flow

This workflow involves only one step and helps in optimization. You cannot roll back as the upgrade is committed automatically.

For more information about ISSU release support and recommended releases, see Technical References → In-Service Software Upgrade (ISSU).

Restrictions and Guidelines for Performing ISSU

- ISSU is supported only on the following SKUs of Cisco Catalyst 3850 Series Switches:
  - C3850-12XS
  - C3850-24XS
  - C3850-48XS
- ISSU is supported only if both the switches in Stackwise Virtual are booted in install mode. (If the chassis is booted in a bundle mode, ISSU is not supported).
- Upgrading hardware and software simultaneously is not supported. Only one upgrade operation can be performed at a time.
- We recommend that upgrades are performed during a maintenance window.
- Do not perform any configuration changes while the ISSU process is being performed.
- ISSU is not supported for an upgrade from IOS XE Fuji 16.9.1 to IOS XE Fuji 16.9.2.
- Downgrade with ISSU is not supported.
- While performing ISSU from Cisco IOS XE Fuji 16.9.x to Cisco IOS XE Gibraltar 16.12.x, if **interface-id snmp-if-index**command is not configured with OSPFv3, packet loss can occur. Configure the **interface-id**

snmp-if-index command either during the maintenance window or after isolating the device (by using maintenance mode feature) from the network before doing the ISSU.

Upgrade Software Using 1-Step WorkFlow

Before you begin

- The device must be booted in the install mode.
- Ensure that the SVL link is up.

SUMMARY STEPS

1. enable
2. **install add file** { **ftp:** | **tftp:** | **flash:** | **disk: *.bin** } **actvate issu commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**install add file** { **ftp:** | **tftp:** | **flash:** | **disk: *.bin** } **actvate issu commit**	Automates the sequence of all upgrade procedures that include downloading the images to both the switches and expanding into packages, and upgrading each switch as per the procedure. **Note** This command throws an error if the switch is booted with a bundle image.

Upgrade Software Using 3-Step WorkFlow

Before you begin

- The device must be booted in the install mode.
- Ensure that the SVL link is up.

SUMMARY STEPS

1. enable
2. **install add file** { **ftp:** | **tftp:** | **flash:** | **disk: *.bin** }
3. **install activate issu**
4. **install commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**install add file** { **ftp:** | **tftp:** | **flash:** | **disk: *.bin** } **Example:** `Switch# install add file ftp:file.bin`	This command downloads the image into the bootflash and expands it on both the switches.
Step 3	**install activate issu** **Example:** `Switch# install activate issu`	On executing this command, the following sequence of events occurs: **a.** A rollback timer is started. If the rollback timer expires, the system rolls back to the same state before the start of the ISSU. The rollback timer can be stopped by using the **install abort-timer stop** command. ISSU can be rolled back using **install abort issu** command. **b.** The standby switch is provisioned with the new software and it reloads with the new software version. Next, the active switch is provisioned with the new software and it reloads. The standby switch with the new image now becomes the active switch and the old active switch becomes the standby. **c.** At the end of this procedure, both the switches run with the new software image.
Step 4	**install commit** **Example:** `Switch# install commit`	The **commit** command performs the necessary clean up, enables the new software as permanent (removing the older version of the software) and stops the rollback timer. Any reboot after the commit will boot with new software. **Note** There is no rollback when this command is used.

Feature Information for ISSU

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Release
ISSU on StackWise Virtual	Cicso IOS XE Fuji 16.9.2